Mathematische Formeln im Web

Eine Ausarbeitung im Rahmen der Lehrveranstaltung Multimedia- und Webtechnologien

Lemgo, Januar 2006

Von
Anke Reddehase
Ilonka Herguth
Julia-Mareike Staff

Inhaltsverzeichnis

1 Vorwort

Im Computerzeitalter wird die Publikation mathematischer Inhalte im World Wide Web immer wichtiger, dies bezieht sich zum einen auf Veröffentlichungen von Lehrern, Professoren usw. für ihre Schüler oder Studenten, die diesen über das Internet zugänglich gemacht werden sollen, aber auch auf jegliche Form der Vermittlung von (Fach-)Wissen über das Web, welches mathematische Formeln enthält.
Problem hierbei ist, dass sich mathematische Formeln meist nur unschön und aufwändig in HTML programmiert werden können, da sie meist eine Menge an mathematischen Sonderzeichen enthalten, die in HTML nicht vorhanden sind.
Dementsprechend muss in diesem Zusammenhang auf verschiedene Hilfsmittel zurückgegriffen werden, um mathematische Formeln in HTML, und damit in Webseiten, zu integrieren.
Möglich ist es, die Formeln über CSS (Cascading Style Sheets) in die Seiten zu integrieren, oder aber sie über MathML, den Standard des W3C’s zur Darstellung mathematischer Inalte im Web, einen „Ableger“von XML, in den HTML- Code einzupflegen.
Hilfreich kann auch das Schriftsatzsystem „LaTeX“ sein, das in seiner Syntax weniger komplex ist als MathML, jedoch mit Hilfe verschiedener Tools leicht in MathML umgewandelt werden kann. Dies ist deshalb nötig, weil LaTeX in seiner reinen Form nicht webfähig ist.
Die jeweiligen Möglichkeiten zur Darstellung mathematischer Inhalte sollen im Laufe dieser Arbeit kurz bezüglich Grundgedanke, Aufbau und Handhabung beschrieben und einige hilfreiche Tools für die jeweilige Lösung vorgestellt werden.

2 HTML und CSS (Ilonka Herguth)

2.1 *Kurze Einführung in CSS*

Cascading Style Sheets, Abkürzung „CSS“ ist eine deklarative Stylesheet-Sprache für strukturierte Dokumente (z. B. HTML und XML), die vom World Wide Web Consortium (W3C) spezifiziert wird.

2.1.1 Idee von CSS

Da in einem HTML-Dokument nur begrenzte Gestaltungsmöglichkeiten vorhanden sind, wurde die CSS-Skriptsprache ins Leben gerufen. Hierbei handelt es sich um eine Sprache, die gezielt für die Webseitengestaltung eingesetzt werden kann.

Durch die Trennung von Stil und Inhalt wird das Veröffentlichen und Betreuen von Dokumenten wesentlich vereinfacht. CSS wurde vor allem im Hinblick auf HTML entwickelt, ist aber auch für XML-Dokumente anwendbar. Es sollen Formatvorlagen erstellt werden, ähnlich wie es bei einem Textverarbeitungsprogramm, z.B. Microsoft Office Word, möglich ist. Es handelt sich also um eine Sammlung von Gestaltungsmöglichkeiten zur Festlegung des Aussehens von statischen HTML-Dokumenten. Die Darstellung der textgestaltenden Eigenschaften erfolgt wie bei HTML-Seiten im Browser des Rechners.

Mit CSS können exakte Vorgaben von Formaten gemacht werden. Die gängigsten Elemente sind die Textgestaltung und die Gestaltung von Hintergründen, Tabellen und Formularen, sowie zur Platzierung von Bildern und Texten.

CSS ermöglicht es auch, Inhalte dem jeweiligen Ausgabemedium (z. B. Druck, Projektion, Sprachausgabe etc.) entsprechend abzuändern. Das ist nützlich, um zum Beispiel Weblinks beim Drucken extra aufzuführen und nicht (wie oft bei HTML-Seiten) zu verbergen.

CSS gilt heutzutage als die Standard-Stylesheet-Sprache für das Web.

Im Dezember 1996 wurde die *CSS Level 1 Recommendation* publiziert. Diese wird in aktuellen Browsern mittlerweile nahezu vollständig umgesetzt.

CSS Level 2 (CSS2) wurde im Mai 1998 veröffentlicht. Bis heute wird diese Empfehlung noch nicht hinreichend von allen verbreiteten Browser implementiert. Gegenwärtig ist *CSS Level 3* in der Entwicklung.

2.1.2 Syntax

Die Syntax besteht aus einzelnen Regeln.
<u>Bsp.:</u>

```
Selector { property-A: value-A; property-B: value-B; } /* Kommentar */
```

Das Ganze stellt (ohne den Kommentar) eine Regel dar. In den geschweiften Klammern befindet sich der Deklarationsbereich, hier mit zwei Deklarationen. Jede Regel kann beliebig viele Deklarationen enthalten.

Selector: Auswahl der gewünschten Elemente (p, body, h1, table, …)

Property: festzulegende Eigenschaft (color, border-style, …)

Value: Eigenschaftswert (red, 20pt, …)

2.1.3 Kombination mit HTML

Die am häufigsten vorkommende *Kombination*, nämlich mit HTML oder XHTML, kann an drei Orten geschehen:

Definition im jeweiligen Tag

Beispielquelltext: style steht für Stilvorlagen

```
<p style="font-family: Verdana; color: red;">
 Heute ist schlechtes Wetter! </p>
```

Ergebnis im Browser:
Heute ist schlechtes Wetter!

Definition im Head einer HTML-Datei

Beispielquelltext: style steht für Stilvorlage, CSS-Stilvorlagen (“text/css”)

```
<html>
 <head>
 <title> 2. CSS im Head </title>

 <style type="text/css">
 p     {
      color: green;                // grüne Schriftfarbe
       font-size: 16pt;              // Schriftgröße: 16 Punkte
       font-family: „Comic Sans MS“; // Schriftart: Comic Sans MS
       ...
       }
 </style>
 </head>

 <body>
 <p> Heute ist schlechtes Wetter! </p>
 </body>
 </html>
```

Ergebnis im Browser:

Heute ist schlechtes Wetter!

Erstellung als externes stylesheet für eine HTML-Datei (`link`-Element)

Die dritte Variante ist in der Praxis am Weitesten verbreitet.

1. Schritt: Erstellung einer CSS-Datei

```
h2      {
      text-align: center;  // zentrierte Textausrichtung
       color: yellow;       // gelbe Schriftfarbe
       background: black;   // schwarzer Hintergrund
```

```
font-size: 16pt;        // Schriftgröße 16 Punkte
}
```

Diese Datei wird unter Dateienendung *.css gespeichert. (Z.B.: cssbeispieldatei.css).

2. Schritt: Verknüpfung des HTML-Dokumentes mit der CSS-Datei

Beispiel HTML-Datei: Link im Head REL=relation

```
<html>
<head>
<title> 3. Separate CSS-Datei </TITLE>

<link REL=stylesheet TYPE="text/css" HREF="cssBeispieldatei.css">
</head>

<body>
<h2> Heute ist schlechtes Wetter! </h2>
</body>
</html>
Alternativ kann zu der Variante mit <link> auch der CSS-spezifische Befehl gewählt
werden:
<style>
@import url("cssBeispieldatei.css");
</style>
```

Ansicht im Browser:

Heute ist schlechtes Wetter!

In der Regel ist die Verwendung eines externen Stylesheets am Sinnvollsten, da dieses die Trennung von Layout und Inhalt ermöglicht.

2.2 Mathematische Formeln mit HTML (und CSS)

Der moderne Trend geht dahin, mathematische markups als spezielle Anwendungen zu betrachten, für die spezifizierte Sprachen wie bspw. MathML genutzt werden.

Es ist jedoch nicht unmöglich mathematische Dokumente in HTML zu schreiben, aber es gibt dabei einige Schwierigkeiten. Die Sprache HTML bietet in gewissem Umfang die Möglichkeit, den Web-Browser zur Darstellung mathematischer Sonderzeichen zu veranlassen ohne Umweg über Browser-Plugins oder die Notwendigkeit, Dokumente in anderen Formaten als HTML downzuloaden. Der code `<font face="Symbol">p</font>` steht bspw. für das Symbol π (pi).

Für Dokumente in denen viele mathematische Notationen gebraucht werden gibt es praktischere Formate als HTML, wie z. B. PDF, PostScript, Tex Format, ...
Typischerweise kann ein *abstract* („Kurzreferat“) nur mit einer überschaubaren Menge an mathematischen Notationen geschrieben werden. Allerdings ist die HTML Version für alle zugänglich und zugleich ist sie auch „Futter“ für Suchmaschinen.

Manchmal ist es das Beste mathematische Ausdrücke als linearisierte Notationen darzustellen.
Dies bedeutet bspw.:

- Anstatt zu versuchen die Quadratwurzel von 2 auf einem normalen mathematischen Weg zu präsentieren, ist es möglich nur *sqrt(2)* zu schreiben.

- Für Quotienten würde man Notationen wie (a+b)/(a-b) benutzen.
- Anstatt Exponenten könnten Notationen wie a**b oder a^b benutzt werden.

Je nachdem, welche Personengruppe angesprochen werden soll, ist es evtl. notwendig solche Notationen zu erklären. Normalerweise sollte das erklärt werden, was nicht der normalen Schulmathematik entspricht.
In einfachen Texten wird x als Multiplikationssymbol benutzt und nicht auf das *Symbol zurückgegriffen.
Als eine gute Idee wird das Nutzen von verschiedenen Typen von Klammern angesehen, bspw. innen ( ), danach [ ], außen { }
Bsp.: `[ (x + y) ^ (1/3) ]  /z`
Eine solche Methode macht das Dokument für alle, die die notwendigen mathematischen Voraussetzungen haben, virtuell verfügbar, wenn man das Schriftzeichen Repertoire nach ISO Lateinisch-1 limitiert.

HTML ist in der Lage, Sonderzeichen auf der Basis verschiedener Standards zu integrieren. So können z.B. direkt die Ziffern-Codes von ISO-8859 oder Unicode, aber auch deren verbale Umschreibungen in Form von Entities dargestellt werden. Deren Präsentation ist allerdings oft vom System, insbesondere von den installierten fonts abhängig. Die Darstellung von Formeln ist über spezielle HTML-Tags, die z.Z. kaum ein Browser implementiert hat, oder über die Einbindung von Grafiken möglich. Über Links lassen sich beliebige Dokumenttypen einbinden, wobei deren Darstellbarkeit von der installierten Browser-Software abhängig ist.

Formeln als Bilder
Eine Möglichkeit ist es, Bilder aus Formeln zu erzeugen. Hierfür gibt es mathematische Programme mit grafischen Darstellungsoptionen, wie z.B. *Mathematica.* Dieses ermöglicht die visuelle Darstellung von Formeln als Bilder.
Siehe hierzu auch Abschnitt 2.4.
Generell kann gesagt werden, dass je komplizierter die Formeln sind, die präsentiert werden müssen, um so ernsthafter sollte das Nutzen von Bildern für diese Absicht berücksichtigt werden, auch wenn dies diverse Nachteile aufweist.

Nutzen von „normalen" Schriftzeichen
Es gibt Browser die, mit einigen Schriftzeichen von ISO Latinisch-1 Probleme haben.
Diese Schriftzeichen beinhalten die hochgestellten Zahlen ^{123}, gewöhnliche Brüche wie ¼, ½, ¾, und das Multiplikationszeichen ×. Basiszeichen wie z.B. das Pluszeichen (+) bereiten keine Probleme. Probleme gibt es dagegen beim Minuszeichen (-), da dieses auch als Bindestrich benutzt werden kann. Es könnte dabei zu Zeilenbruch-Problemen kommen, da die Browser beim Bindestrich brechen, nicht aber beim Minuszeichen. Als Multiplikationssymbol kann entweder das × verwendet werden ($a \times b$) oder der Mittel-Punkt ($a \cdot b$).
Wird das ×-Symbol benutzt, sollte zwischen den Zahlen ein Dezimalpunkt gesetzt werden, beim · - Symbol sollte hingegen nach ISO 31-0 Standard ein Komma benutzt werden.
Bsp.:
```
4.2&times;10<sup>15</sup>
4,2&middot;10<sup>15</sup>
```

Das *-Zeichen wird eher selten benutzt, da es ein Operationssymbol ist.
Bei der Division wird / bevorzugt, oder in einigen Fällen ÷.
Operatoren wie größer oder kleiner als <> sind in HTML tag- Abgrenzungen notwendig, daher sollten die Notationen `<` und `>` genutzt werden.
Ungleichheitssymbole wie ≠; ≤; ≥ sind problematisch, da sie nicht zu ISO Latinisch-1 gehören. Wenn sie trotz der Probleme benutzt werden, dann sollten sie als nummerische Referenzen verwendet werden: `≠` (≠) und `≤` (≤) und `≥` (≥) oder es werden Vertreter wie: =/ ; <= und >= gewählt. Das Verwenden von solchen Nichtgleichheitszeichen ist sehr schwierig, da es keine weltweit anerkannten Vertreter gibt.

Spezielle markups oder CSS Regeln können den Browser dazu bringen einen kursiven font zu benutzen, wie bspw. innerhalb eines `em` Elements (welches typischerweise kursiv dargestellt wird).
Wenn das Symbol in dem Text erscheint, der von dem markup oder den CSS Regeln betroffen ist, sollte in HTML das Folgende genutzt werden:

```
<em>The measured time was only 42 <span class="unit">s</span>.</em>
```

und in CSS:

```
.unit { font-style: normal; }
```

Das Problem der der Trennung
Ein weiteres Problem ist das Trennen von sehr langen Zahlen bzw. Zifferngruppen, da 1,005 und 1.005 in verschiedenen Ländern eine unterschiedliche Bedeutung haben. Daher ist es am sichersten Zahlen mit vielen Ziffern durch *kleine* Abstände zu trennen, nie durch Punkt oder Komma (ISO 31-0 Standard), Bsp. 1 000 000 000 000.
Es können normale Schriftzeichen für Abstände benutzt und einige leichte CSS-Regeln spezifiziert werden, die reduzierte Abstände zwischen den Wörtern vorschlagen, indem für die `word-spacing` Eigenschaft ein negativer Wert benutzt wird. Die Eigenschaft spezifiziert einen zusätzlichen Abstand zu dem Leerzeichen, d.h. ein negativer Wert bedeutet die Reduzierung des Abstandes.

`word-spacing: -0.07em` liefert ein gutes Resultat für Abstände zwischen Zifferngruppen. Dies bedeutet eine Reduzierung des normalen Abstandes um 7%.
So sind die Zifferngruppen getrennt sichtbar und der Abstand wirkt auch nicht störend.
Voraussetzung hierfür ist, dass der Browser diesen Teil der CSS-Spezifikation unterstützt.

Bsp.:

```
<span class="number">123 456 789 000 000 000</span>
```

Style Sheet:

```
.number { word-spacing: -0.07em; white-space: nowrap; }
```

Wird bspw. das Programm TeX genutzt, so werden die Abstände sorgfältig kontrolliert, indem vorteilhafte Werkzeuge und Techniken benutzt werden, was eine hohe Qualität mit sich bringt. Es kann nicht erwartet werden, das Gleiche mit HTML und CSS zu erreichen, aber es wird dennoch auf eine vernünftige Qualität abgezielt.

Ein Ausdruck wie "$a + b$" wird am Besten in HTML geschrieben, so dass dort Abstände um den Operator „+" sind. Dies gibt mehr Flexibilität, weil das `word-spacing` in einem einfachen Weg genutzt werden kann.

Bsp.: Ausdruck mit verschiedenen Operatoren unter verschiedenen style sheets

unstyled	a + (b × c) / (x + y + z)
Word-spacing:-0.07em	a + (b × c) / (x + y + z)
Word-spacing:-0.2em	a + (b × c) / (x + y + z)

Das Benutzen von speziellen Schriftzeichen
In mathematischen Notationen werden oft spezielle Symbole gebraucht, wie bspw. griechische Buchstaben oder das Symbol für Unendlichkeit (∞). Im Prinzip kann das vollständige Unicode-Schriftzeichen Repertoire in HTML benutzt werden. In der Praxis ist jedoch die Browserunterstützung limitiert.

Für Schriftzeichen, die nicht zuverlässig genug dargestellt werden können aber dennoch gebraucht werden, können verschiedene Vertreter verwendet werden. Eine vorteil-

hafte und oft völlig ausreichende Methode ist es, Wörter oder Abkürzungen zu benutzen, bspw. „infinity“ oder „inf“ statt ∞. Logische Verbindungen wie „and“ und „or“ können sicher dargestellt werden, wenn /\ and \/ benutzt wird (vielleicht in reduzierter Größe ∧ ∨).
Tatsächlich hat der umgekehrte Schrägstrich \ keine feste Bedeutung, die von einer solchen Nutzung gestört würde und ursprünglich wurde er für die Nutzung in solchen Notationen eingeführt.
Es ist allerdings zu beachten, dass die Unicode Schriftzeichen für Verbindungen ∧ und ∨ gegenwärtig schlecht unterstützt werden.
Ein allgemeiner Vertreter für Pfeilsymbole ist der Gebrauch von Schriftzeichenpaaren wie <- oder <= , längere Zeichenfolgen wie ---> können die Bedeutung noch offensichtlicher machen.
Wenn Notationen wie /\ oder ----> versuchen die Gestalt der realen Schriftzeichen, für die sie stehen, zu interpretieren, kann es nützlich sein einige spezielle fonts vorzuschlagen, indem man das `font-face` markup in HTML oder die `font-family` (Schriftart) Eigenschaft in CSS benutzt.

style sheet:

```
.logop { font-family: "Times New Roman", serif }
.arrow { font-family: Tahoma, Symbol, monospace }
.darrow { font-family: "Times New Roman", serif }
```

HTML markup:

```
<span class="logop"><nobr>/\<nobr><span>
<span class="arrow"><nobr>--></nobr><span>
<span class="darrow">==><span>
```

Browser: /\ --> ==>

Will man Formeln wie $E = mc^2$ hervorheben benutzt man ein `big` markup oder die `font-size` Eigenschaft in CSS zusätzlich zu dem logischen `em` oder dem `strong` markup.
Wenn ein kursives Schriftzeichen einem nicht-kursiven folgt kann es passieren, das es im Browser oft zu dicht aneinander dargestellt wird.

Bsp.: *f*(0) dann evtl. no-break space Schriftzeichen dazwischen *f* (0)

Oder es wird ein markup wie <i>f</i> (0) benutzt.

<u>Brüche</u>
Für Brüche wie 6/7 ist die allgemeine linearisierte Notation gewöhnlich die Beste, insbesondere innerhalb eines Textes. Der Versuch echte zweidimensionale Brüche zu erzeugen, würde relativ komplizierte Techniken verlangen. Daher wird häufig auf die „pseudo-Zweidimensionalität“ zurückgegriffen.
Es kann eine linearisierte Notation mit `sup` markup für den Zähler und `sub` markup für den Nenner benutzt werden. Das Hauptproblem liegt darin, dass ein Ausdruck wie $^5/_8$ aufgrund der schlechten Implementationsqualität des `sup` und `sub` dazu tendiert, unebene Zeilenabstände zu verursachen.
Verschiedene Darstellungen für $^5/_8$:

Approach	HTML notation	Appearance
Linear notation	5/8	5/8 text
Special character	⅝	⅝ text
sup and sub	⁵/₈	5/8 text
Fraction slash	5⁄8	5⁄8 text
Fraction slash and CSS	5⁄8	5⁄8 text

Unterstriche / Überstriche

Das Überstreichen wird in der Mathematik öfters benutzt, z.B. um den Durchschnitt aufzuzeigen. In HTML ist ein Überstreichen nicht möglich.
In style sheets kann überstrichen werden, indem die Erklärung `text-decoration: overline` vorgeschlagen wird. Für das Überstreichen eines einzelnen Symbols könnte ein eingebettetes style sheet wie Folgendes benutzt werden:

```
<b style="text-decoration:overline"><i>x</i></b>.
```

Für Wurzeln $\sqrt{}(a^2+b^2)$ kann das Überstreichen vorgeschlagen werden um zu verdeutlichen was unter die Wurzel gehört.
Hierzu werden einfache markups, bei denen die Ausdrücke in Klammern zwischen `<span style="text-decoration:overline">` und `</span>` eingeschlossen sind, benötigt. Es ist zu beachten, dass durch den Gebrauch von `sup` als Exponenten unter der Wurzel die Darstellung auf vielen Browsern verzerrt wird.
Außerdem könnte CSS benutzt werden, um reduzierte Abstände zwischen diesen Schriftzeichen vorzuschlagen.
HTML markup:

```
<span class="radic"><sup><var>n</var></sup>&#8730;</span><span
class="radicand"><var>x</var></span>
```

CSS:

```
.radic {letter-spacing:-0.15em; }
.radicand {text-decoration:overline;  }
```

Browser: $\sqrt[n]{x}$.

Felder und Tabellen

Nummerische Daten sollten normalerweise rechtsbündig dargestellt sein, weshalb häufig `align="right"` gebraucht wird.
Oft ist es wünschenswert nummerische Daten am Dezimalpunkt auszurichten, aber das ist in der Praxis nicht möglich. Statt dessen, müssen einige Tricks gebraucht werden, wie z.B. das Nutzen von monospace fonts und right padding mit no-break spaces, so dass die Inhalte einer Spalte die gleiche Anzahl an Schriftzeichen rechts des Dezimalpunktes haben. Für eine Matrix ist das Nutzen von großen runden Klammern rundherum eine herkömmliche Notation in der Mathematik. Dies wird in HTML weitaus schwieriger und würde nur für kleine Matrizen gut funktionieren.
Vielleicht ist es das Beste eine andere Darstellung zu nutzen, die den Matrizen ein charakteristisch geeignetes Erscheinen gibt, wie bspw. einen speziellen aber nicht zu auffallenden Hintergrund für die Zellen. Für eine Tabelle, die eine Matrix darstellt, könnte geschrieben werden:
`<table class="matrix">` und eine style sheet Regel wie Folgende wird mit einbezogen:

```
table.matrix td { background: #fda none; color:#000; }
```

Browser:

A =

x-y	*-a*	42
C	*a*b*	*c*

Die obige Tabelle, die eine Matrix darstellt, ist in eine äußere Tabelle eingebettet, so dass ein Symbol mit der Tabelle assoziiert werden kann.
Ähnliche Techniken können z.B. benutzt werden, wenn die Summe von Tabellen dargestellt werden soll; es würde eine äußere ein Reihen(row)-Tabelle geschrieben, die in ihren Zellen die Matrizen enthält und ein Plus-Zeichen in einer eigenen Zelle dazwischen.
Bsp.:

$$\frac{15}{37} + \frac{22}{22} = \frac{37}{59}$$

Das Problem hochgestellter Positionen

Ein Beispiel indem auch spezielle Schriftzeichen benutzt werden:

```
&#8721;<sub><var>i</var>=0</sub><sup>&#8734;</sup>
<var>x<sub>i</sub></var>
```

$\sum_{i=0}{}^{\infty} x_i$

Dies bedeutet die Summe x_i von i=0 bis unendlich.
Das Problem in den meisten Browsern ist, dass das Unendlichkeitssymbol nicht über i=0, sondern rechts davon auftaucht. Im schlimmsten Fall könnte der Betrachter das obere Limit als ein Exponent des unteren Limits missverstehen. So ist es besser überhaupt keine Hochstellung zu benutzen, aber die Limits tiefer zu stellen, z.B. i=0,…,∞.

Folgendes Erscheinen: $\sum_{i=0,\ldots,\infty} x_i$.

Im Allgemeinen werden spezielle Symbole leicht unleserlich wenn die `font-size` reduziert wird, daher sollte die `font-size` größer als normal gewählt werden.
Es gibt verschiedene Weisen einen Summenausdruck darzustellen. Generell führen sie zu komplizierten Konstrukten und die Komplexität kann Probleme auf verschiedenen Browsern aufwerfen.

Es taucht das Problem auf, dass `sub` und `sup` schnell zu unebenen bzw. ungleichen Zeilenabstände führt. Eine mögliche Lösung wäre, die `sup` oder `sub` Elemente in `small` markups zu setzen.
Bsp.: `10<small><sup>9</sup></small>`.
Jedoch könnte eine Reduktion der Größe den Text, besonders auf Browsern die bereits reduzierte Größe für `sub` und `sup` nutzen, schnell unlesbar machen. Daher ist es das Beste CSS zu benutzen, um eine vernünftige Größenreduktion vorzuschlagen. Am Besten ist es ein sans-serif font in diesem Zusammenhang zu verwenden, da solche fonts gewöhnlich lesbarer als serif fonts sind wenn die font size klein ist.
Bsp. CSS Regel:

```
sub, sup { font-family: Arial, Helvetica, sans-serif;
           font-size: 65%; }
```

Browser: 10^9 oder A^T .Der Zeilenabstand vergrößert sich nicht oder nicht auffällig.
Das Problem der ungleichen Zeilenabstände kann meistens vermieden werden, wenn `sub` und `sup` nur in Formeln benutzt wird, die als separate „Blöcke“ dargestellt werden, nicht im fließenden Text.

Spezielle Probleme mit Exponenten

Aber es gibt noch ein anderes Problem, insbesondere wenn `sup` als Exponent gebraucht wird. Denn es macht wirklich einen Unterschied ob 10^9 was bedeuten soll 10 mit der Potenz 9, tatsächlich als 109 angezeigt wird. Es ist richtig, dass die meisten Browser die Eigenschaft `sup` übergeben, aber es scheitert oft daran es schön zu übertragen. Das bedeutet, dass eine Exponent an einem Exponent in der graphischen Darstellung verloren gehen könnte.
z.B. übergeben einige Versionen des IE `a<sup>b<sup>c</sup></sup>` genauso wie `a<sup>bc</sup>`.
Bsp. für das Erscheinen im Browser: a^{b^c} (c sollte ein sup von b sein).
Das Vorangegangene kann das übertragene `sup` als ziemlich kleiner font zeigen. Wenn übertragene `sup` wie in a^{bc} oder $e^{f(x)n}$ gebraucht werden, ist es vielleicht das Beste die `font-size` nur auf z.B. 93% zu reduzieren. Das könnte natürlich wieder zu unebenen Zeilenabständen

führen. Eine Methode ist es, die Eigenschaft der Zeilenhöhe (in CSS) auf einen relativ hohen Wert wie 1.6 zu setzen.
Außerdem könnte es eine gute Idee sein, die Zeilenhöhe global (für das Body Element) auf einen Wert wie 1.3 zu setzen, wenn das bei einigen kleineren Problemen wie unebene Zeilenabstände hilft und die Lesbarkeit im Allgemeinen verbessert.

Probleme der Hochstellung vermeiden
Potenzierung wird öfters durch das Nutzen von Vertreter-Notationen, welche Exponentialoperatoren wie ** oder ^ benutzen, ausgedrückt. Es könnten in einigen Fällen auch Funktions-Notationen wie exp(x) anstatt e^x verwendet werden.
Für eine generelle Potenz könnte eine Notation wie Potenz (a,b) gebraucht werden, aber der Autor sollte angeben ob es die b^{th} (b-te) Potenz von a oder die a^{th}
(a-te) Potenz von b ist. Für Zahlen könnte der Gebrauch von 42.5×10^9 durch eine Notation wie 42.5E9 ersetzt werden.

Gleichungszahlen/-mengen
In der Mathematik ist es weit verbreitet, Gleichungen zu nummerieren und die Zahl rechts einer Gleichung in Klammern zu setzen.
Bsp.:

$$(a + b)^2 = a^2 + 2ab + b^2 \qquad \textbf{(42)}$$

$$e^x \approx 1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120 + x^6/720 + x^7/5040 + x^8/40320 + x^9/362880 + x^{10}/3628800 \qquad \textbf{(43)}$$

Verschiedene Methoden werden vorgeschlagen, um ein solches Layout zu erreichen indem CSS benutzt wird. Einige Methoden funktionieren gut wenn die Gleichung in eine Zeile passt, aber führen zu Verwirrung wenn sie auf zwei oder mehrere Zeilen aufgeteilt ist. Sofern der Wunsch besteht Seiten zu erstellen, die sich an schwankende Planungsweiten („fließendes Design“) anpassen, ist die praktische Lösung eine einfache Tabelle:

```
<table class="eq" summary="Equation and its number." width="100%">
<tr>
<td>the equation</td>
<th align="right" valign="bottom">(number)</th>
</tr>
</table>
```

Man kann sich von den Darstellungsattributen width, align, und valign befreien, indem die entsprechende CSS Eigenschaften benutzt wird.
Wenn verhindert werden soll, dass die Gleichungsnummer fett erscheint, sollte das `th` Element durch das `td` Element ersetzt oder die `font-weight` Eigenschaft in CSS benutzt werden.

style sheet:

```
.eq { width: 100%; }
.eq th { text-align: right;
         vertical-align: bottom;
         font-weight: normal; }
```

Die folgende Gleichung mit solch einem style sheet formatiert:

$$(p + q)(r + s) = (p + q)r + (p + q)s = pr + qr + ps + qs \qquad (44)$$

Eine Möglichkeit einen Ausdruck zweidimensional erscheinen zu lassen, ist das Nutzen einer Tabelle:

```
<table cellspacing="0" cellpadding="0">
<tr><td align="center"><i>x</i></td></tr>
<tr><td valign="middle"><img src="1px.gif" alt="divided by"
 width="35" height="1"></td></tr>
```

```
<tr><td align="center"><i>a</i> - <i>b</i></td></tr>
</table>
```

Erscheinen im Browser:

$$\frac{x}{a - b}$$

Das einzige, was wirklich ein Trick ist, ist der Gebrauch eines 1px.gif mit dem `width` Attribut zu einer horizontalen Linie ausgedehnt. Eine Weite von 35 Pixeln ist nur eine Schätzung, sie scheint kein guter Weg zu sein, die Weite in vernünftigere Einheiten zu spezifizieren. (Es könnte die `em` Einheit in CSS benutzt werden, aber sie hat kein definiertes Verhältnis zu der Breite der Schriftzeichen; als eine sehr unsichere Faustregel ist `1em` zwei Schriftzeichen breit).

Indem man ein etwas erfinderisches Tabellenmarkup benutzt, könnte das Folgende erzielt werden:

$$\int^{b} \frac{f(x)}{1 + x}\, d x$$

(a+b)/x könnte zweidimensional dargestellt werden, indem style sheets, speziell die `display` Eigenschaft in CSS1, benutzt wird. Es wird mit leichten linearisierten Notationen wie (a+b)/x begonnen. Klammern und Slash für `span` Element und `span` für Zähler und Nenner. Dann werden Klassennamen benutzt und den `span` Elementen zugewiesen. Es wird die Unterdrückung des Displays von den Schriftzeichen in CSS vorgeschlagen, die Zähler und Nenner als einen Block darstellen und den Zähler unterstreichen.

Markup:

```
<span class="nom"><span class="lin">(</span><i>a</i> -
<i>b</i><span class="lin">)</span></span><span
class="lin">/</span>
<span class="den"><i>x</i></span>
```

style sheet:

```
.lin { display: none; }
.den, .nom { display: block;  width:100%; text-align:center }
.nom { text-decoration: underline; }
```

Browser:

$\underline{a - b}\ x$

Und es setzt es in nicht CSS-fähigen Browsern auf $(a - b)/x$ herab.

Es ist zu beachten, dass der Nenner überstrichen werden könnte, wenn der Nenner breiter ist als der Zähler, statt den Zähler zu unterstreichen. Unglücklicherweise unterstützt Netscape 4 das Überstreichen nicht.
In HTML kann eine Vielzahl von mathematischen Symbolen und einigen anderen Basisnotationen genutzt werden, aber zur Zeit mit Zugangsproblemen, die die Benutzer lösen müssen. Daher ist es besser Bilder zu verwenden.

2.3 Integration einzelner Formeln in HTML-Seiten

Wie schon erwähnt, gibt es verschiedene Bemühungen der Darstellung mathematischer Symbole und Formeln in HTML-Seiten.

Das derzeit übliche Verfahren zur Bereitstellung mathematischer Texte im Internet ist die Dateiform im TeX-Ausgabe- (*.dvi) bzw. PostScript®-Format (*.ps). In sehr eingeschränktem Umfang können mathematische Sachverhalte aber auch mit den Mitteln von HTML realisiert werden.

Nachstehend sollen einige Aspekte eines weiteren, aber auch seltener praktizierten, Ansatzes diskutiert werden. Hierbei ist das eigentliche Anliegen nicht die Bereitstellung ganzer Dokumente, sondern die Integration von einzelnen Formeln in mit HTML gestalteten Seiten. Dieser Ansatz kann wie folgt zusammengefasst werden:

- Der gewöhnliche Fließtext wird mit den Standardmitteln der Beschreibungssprache HTML erstellt.
- Formeln werden als Grafiken mit der in Windows-Anwendungen üblichen Auflösung platziert.

Nachteile:

- Während die Schriftgröße seitens des Lesers beliebig einstellbar sein sollte, ist die Größe einer Grafik fest vorgegeben. Damit sind ästhetische Konflikte vorprogrammiert.
- Das vorstehende Argument gilt bereits für den Fall in separaten Zeilen hervorgehobener Formeln und verschärft sich bei Formeln und mathematischen Symbolen, die im Fließtext verwendet werden sollen.
- Durch viele Grafiken wird auch das Datenvolumen unverhältnismäßig erhöht.

Vorteile:

- Es stehen alle Möglichkeiten des Systems zur Verfügung, das für die grafische Erzeugung der Formeln eingesetzt wird.
- Alle Schriftarten, die in diesem System zur Verfügung stehen, können zur Gestaltung herangezogen werden, ohne dass diese im Zielsystem (Browser) verfügbar sein müssen.
- Die Grafiken können mit Bildbearbeitungsprogrammen nach den eigenen Vorstellungen nachbearbeitet werden.
- Eine Speicherung im binären GIF-Bildformat (Graphic Interchange Format) benötigt vergleichsweise wenig Speicherkapazität.

Abschließend bleibt auch festzustellen, dass die Verwendung dieser Technik für mathematische Symbole und Formeln im Fließtext nur bedingt geeignet und aufwendig erscheint. Zum einen stellt HTML nur wenige Möglichkeiten der Ausrichtung der Grafik in Bezug zum Text, in dem sie erscheinen soll, zur Verfügung, zum anderen - teilweise aus diesem Umstand folgend - ist die Dimensionierung der Grafik unverhältnismäßig kompliziert. Im Fließtext eingefügte Grafiken führen auch zu unregelmäßigen Zeilenabständen.

2.4 Weitere Möglichkeiten zur Formelerstellung

Eine direkte Umsetzung mathematischer Formeln in HTML führt auf Konstrukte wie etwa

```
<DIV ALIGN="CENTER">
|P> = a<sub>1</sub>|1>+a<sub>2</sub>|2>+...+a<sub>N</sub>|N>
</DIV><BR>
<DIV ALIGN="CENTER">
a<sub>n<sub>1</sub>n<sub>2</sub></sub> =
Ae<sup>i(k<sub>1</sub>n<sub>1</sub>+k<sub>2</sub>n<sub>2</sub>)</sup>
+ Be<sup>i(k<sub>1</sub>n<sub>1</sub>+k<sub>2</sub>n<sub>2</sub>)</sup>
</DIV>.
```

Auf der fertigen Webseite erscheint dies dann so :

$|P> = a_1|1>+a_2|2>+...+a_N|N>$

$a_{n1n2} = Ae^{i(k_1 n_1 + k_2 n_2)} + Be^{i(k_1 n_1 + k_2 n_2)}$

Diese Darstellung der mathematischen Formeln ist so schlecht leserlich, dass längere Textpassagen mit Formeln den Leser schnell ermüden.

Besser leserliche Formeln lassen sich bekanntlich mit TeX oder LATeX erzeugen. Um diese auch in HTML-Dokumenten nutzen zu können wurde das Werkzeug latex2html entwickelt, welches die mathematischen Formeln zunächst mit LATeX zu kleinen Bildchen verarbeitet, und dann das Ergebnis als Bild in ein HTML-Dokument einbindet. Als Ergebnis erhält man für die obigen Beispiele dann

$$|P\rangle = a_1 \,|1\rangle + a_2 \,|2\rangle + ... + a_N \,|N\rangle$$

und

$$a_{n_1 n_2} = A\mathrm{e}^{\mathrm{i}(k_1 n_1 + k_2 n_2)} + B\mathrm{e}^{\mathrm{i}(k_1 n_1 + k_2 n_2)}.$$

Die Technik: Das *img* Element
Beispiel:
Ein leichtes markup wurde in TeX benutzt, davon wurde eine GIF Bilddatei erzeugt und in HTML mit folgendem markup eingebunden:

```
<p><strong>Assignment 42</strong>. Compute
<img src="integral.gif" align="middle" alt=
"the integral of exp(x**2) for x from 0 to infinity"></p>
```

Browser:

Assignment 42. Compute $\int_0^\infty e^{x^2} dx$

Es ergeben sich weitere Möglichkeiten, wenn andere Programme oder Sprachen zur Erstellung von Formeln benutzt werden. Ist ein Dokument beispielsweise in LaTeX geschrieben, so kann es mit Hilfe des Programms TTH: a TeX to HTML translator in HTML-Code umgewandelt werden. Dieses benützt dieselben Sonderzeichen wie HTML. Für aufwendige Formeln werden kunstvolle Tabellenkonstruktionen angewandt, und Wünschenswertes kann dennoch meistens nicht dargestellt werden.

HTML sieht keine Elemente zur Beschreibung von Formeln vor. So werden Formeln als Grafiken abgespeichert und eingebunden, mit komplizierten Tabellenstrukturen und Symbolschriften nachgebaut, und jede dieser Methoden ist gleichermaßen aufwendig wie ineffektiv. Die Lösung dieses Problems bietet Xhtml in Verbindung mit MathML.

2.5 MathType-Formeln über das Internet verbreiten

2.5.1 Was ist MathType?

MathType™ wird als die Profiversion des Formel-Editors bezeichnet.

MathType ist ein leistungsstarkes, interaktives Zusatzprogramm für Windows und Macintosh, mit dem mathematische Ausdrücke in Textverarbeitungs- und DTP-Dokumenten, in Webseiten, in Präsentationen sowie in TeX-, LaTeX- und MathML-Dokumenten erzeugt werden können.

MathType verwendet eigene Schriftarten, die auf Systemen, auf denen MathType nicht installiert ist, nicht vorhanden sind. Für eine einwandfreie Bildschirmdarstellung der Formeln sind diese Schriftarten jedoch nötig.

Zur Lösung des Problems gibt es verschiedene, im Folgenden erläuterte, Methoden:

HTML-Konverter
In vielen Textverarbeitungsanwendungen (z.B. Microsoft Word, Corel WordPerfect etc.) lassen sich Dokumente inzwischen im HTML-Format speichern und so im Web-Browser direkt anzeigen.

Vorteile

- Dokumente lassen sich ohne große HTML-Kenntnisse schnell und einfach in das HTML-Format konvertieren.
- Formeln werden automatisch in GIF-Dateien konvertiert.

Nachteile

- Die vertikale Ausrichtung der Formel zum Text wird nicht beibehalten (Inline-Formeln werden im Text zu hoch oder zu tief platziert).
- Die GIF-Konvertierung führt zuweilen zu verzerrten Resultaten (Probleme mit Abständen und der Ersetzung von Schriftarten). Dies liegt an der Konvertierungsfunktion der Textverarbeitung; mit direkt in MathType erstellten GIF-Dateien verhält es sich anders.
- Je nach Auflösung des jeweils verwendeten Bildschirms erscheinen GIFs entweder zu groß oder zu klein
- Formeln lassen sich nicht mehr bearbeiten, da durch den GIF-Konvertierungsvorgang sämtliche Informationen aus der Datei entfernt werden, die zur Bearbeitung des Ausdrucks in MathType nötig sind.
- GIF-Dateien haben eine niedrige Auflösung, was sich in schlechter Druckqualität äußert.

Wenn hauptsächlich Wert darauf gelegt wird, dass die Formeln möglichst SCHNELL in möglichst vielen Browsern angezeigt werden, ist diese Option die Richtige. Allerdings besteht der Nachteil der Geschwindigkeit darin, dass das Endergebnis sich nicht mehr formatieren lässt. Mit dieser Methode kann zwar jeder Anwender HTML-Dokumente anfertigen und ins Internet stellen, auf Grund der durch mangelnde HTML-Unterstützung bedingten Formatierungsveränderungen (besonders was die vertikale Ausrichtung angeht) ist diese Option jedoch nur für Anwender geeignet, die ihre Dokumente mit Anzeigeformeln, also in separaten Textabsätzen platzierten Ausdrücken, versehen.

GIF, HTML und CSS

Eine weitere Option zur Webseitenerstellung besteht darin, das Dokument mit einem Texteditor direkt in HTML zu erstellen. In MathType lassen sich Formeln als GIFs speichern; gleichzeitig werden ein HTML-<IMG>-Tag und die Attribute erstellt, mit deren Hilfe die GIFs in das HTML-Dokument eingefügt werden können. Wenn HTML zusammen mit CSS verwendet wird, lässt sich das Problem der vertikalen Positionierung direkt angehen. In MathType gibt es die Möglichkeit, verschiedene Tag-Informationen einzuschließen, die die Erstellung der
CSS-Tag-Information unterstützen.
Ein mögliches Beispiel hierfür wäre:

```
<IMG SRC=INTEGRATION.GIF STYLE="POSITION:RELATIVE; TOP:21.7PT" HEIGHT=88 WIDTH=256>
```

Vorteile

- Die GIF-Konvertierung verläuft stets korrekt (keine Probleme mit Abständen und der Ersetzung von Schriftarten).
- In MathType erstellte GIFs können auch weiter in MathType bearbeitet werden.
- Die Formeln werden in jedem Browser und auf jeder Plattform korrekt angezeigt, da sie als Grafikdatei gespeichert sind.
- Über CSS ist eine präzise vertikale Ausrichtung möglich.

Nachteile

- Sie benötigen gute Kenntnisse der HTML- und CSS-Syntax.
- Wenn Sie CSS verwenden, ist die Anzeige auf CSS-fähige Browser beschränkt.
- Je nach Auflösung, mit der das GIF erstellt wurde, und dem jeweils verwendeten Bildschirm erscheint die Formel entweder zu groß oder zu klein
- GIF-Dateien haben eine niedrige Auflösung, was sich in schlechter Druckqualität äußert.

Mit guten HTML- und CSS-Kenntnissen lassen sich auf diese Weise Formeln annähernd originalgetreu in Webdokumenten darstellen. Für HTML-Kundige mag diese Methode sicher gut geeignet sein; Anwender jedoch, die ihre Formeln nur einfach und schnell über eine Textverarbeitung ins Internet stellen möchten, haben nicht viel davon. Außerdem löst dieser Ansatz nicht das Problem der Bildauflösung.

2.6 Beispiel für das Erstellen von Formeln mit CSS

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head><title>Beispiel</title>

<style type="text/css">
body     {
         background-color:#dedeed;
             }

div.generell {
  white-space: nowrap;
  font-family: sans-serif;
  font-style: bolder;
  font-size: 24pt;
  color: navy;
  margin-left: 3em;
  margin-right: 1em;
  float:middle
  }
```

```
span.f {
  display:-moz-inline-box;
  -moz-box-orient:vertical;
  display:inline-block;
  vertical-align:middle;
  margin:0 0.2em;
  font-size: 24pt;
  }

span.bruchstrichoben {
  display: block;
text-align:center;
  margin:0 0.2em;
border-bottom:solid 1px navy;
}

span.bruchstrichmitte {
  display: block;
text-align:center;
  margin:0 0.2em;
border-bottom:solid 3px navy;
}

span.bruchstrichunten {
  display: block;
 text-align:center;
  margin:0 0.2em;
}

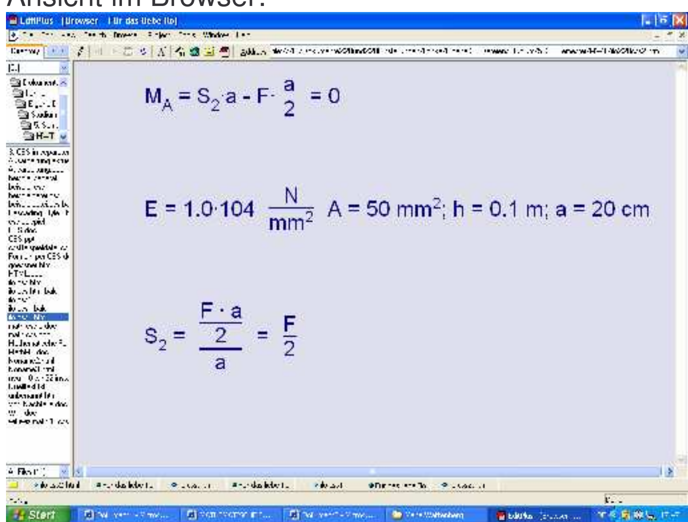
</style></head>
<body>
<div class="generell">M<sub>A</sub> = S<sub>2</sub>·a - F·<span class="f"><span
class="bruchstrichoben">a</span><span class="bruchstrichunten">2</span></span> =
0</div>
<br><br><br>

<div class="generell">E = 1.0·104 <span class="f"><span
class="bruchstrichoben">N</span><span
class="bruchstrichunten">mm<sup>2</sup></span></span> A = 50 mm<sup>2</sup>; h =
0.1 m; a = 20 cm</div>
<br><br><br>

<div class="generell">S<sub>2</sub> = <span class="f"><span
class="bruchstrichoben">F · a </span><span class="bruchstrichmitte">2</span><span
class="bruchstrichoben"></span><span class="bruchstrichunten">a</span></span> =
<span class="f"><span class="bruchstrichoben">F</span><span
class="bruchstrichunten">2</span></span></div>

</body></html>
```

Ansicht im Browser:



3. MathML (Anke Reddehase)

3.1 Einführung

Millionen Menschen auf der Erde verwenden die mathematische Standardschreibweise. Sie ist die Sprache der Wissenschaft.

Mathematische Formeln einfach ins Internet zu stellen - diesen Wunsch hegten schon einige Computerbenutzer aus den naturwissenschaftlichen, den mathematischen und den technischen Fachbereichen in der Vergangenheit.

Als 1997 die W3C Math Working Group gegründet wurde, keimte die Hoffnung auf, dass nun endlich ein Standard eingeführt werde, der sich mit dieser Problematik ernsthaft auseinandersetzen und auch eine Lösung bereitstellen sollte.

Einige Jahre zogen seither ins Land und inzwischen sind die Früchte der Working Group reif genug, **MathML** ist das Akronym und steht für **Mathematical Markup Language**.

3.1.1 Ein Blick in die Vergangenheit

Die meisten modernen Textverarbeitungsprogramme von heute unterstützen zwar die Erstellung und Bearbeitung von mathematischen Ausdrücken, Webseitenerstellungsprogramme unterstützten dies jedoch nicht. Der wichtigste Grund dafür ist das HTML (die Definitionssprache für Webseiten) die Darstellung mathematischer Ausdrücke nicht unterstützt.

Früher wurden mathematische Inhalte mühselig als Bilddateien, durch das Hinzufügen von Links auf z.B. GIF-Bilder (Graphics-Interchange-Format, das Standard-Bildformat für Strichzeichnungen), in Internetseiten integriert. Dies hat jedoch minderwertige Druckqualität und lange Downloadzeiten zur Folge und die Formeln werden nicht an Schriftarteneinstellungen im Browser angepasst. Im JPEG-Format ist dies ebenfalls möglich, jedoch eignet sich dies am besten für fotografische Bilder.

Wenn Sie allerdings sehr viele Formeln auf einer Seite darstellen möchten, führt dies natürlich auch zu sehr viel längeren Ladezeiten.

Die Nachteile dieser Vorgehensweise auf einen Blick:

- Schriftarten werden bei der Dokumentenerstellung festgelegt
- GIF Bilder werden als Bitmap gedruckt, Druckqualität ist nicht besonders gut
- Das Herunterladen einer Seite mit mehreren GIF Bildern dauert Verhältnismäßig lang, das Herunterladen jeder Formel stellt eine gesonderte Transaktion mit dem Server dar
- Es ist nicht möglich die Formel an ein anderes Programm zu übertragen und dort zu bearbeiten, da die mathematische Struktur nicht erhalten bleibt

3.2 Mathematikal Markup Language – eine kurze Beschreibung

- Dokumentenformat zu Darstellung von mathematischen Formeln und komplexen Ausdrücken (mathematische Gleichungen aller Schwierigkeitsgrade)
- Durch eine Spezifikation des World Wide Web Consortuims festgelegt (Anfang 2001)
- Basiert auf der Beschreibungssprache XML (einem Nachfolger von HTML)
- Untermenge der Sprache XML, der Gebrauch von XML ist also zwingend
- Grundlage für die Einbindung mathematischer Gleichungen in Webseiten

- Kann als JPEG oder Bitmap Datei in unterschiedlicher Auflösung gespeichert werden
- MathML wurde in der Programmiersprache C++ erstellt
- Technik entspricht konzeptionell derjenigen anderer Markup Languages wie z.B. HTML
- Inhalt lässt sich durch Tags steuern z.B. Wurzelzeichen mit <msqrt>
- Ist besonders dazu geeignet eine große Anzahl an Formeln aufzubereiten, da sehr schnell Anpassungen vorgenommen werden können.
- Ist dazu konzipiert von Software gelesen zu werden
- Kann von Menschen gelesen werden ist jedoch dazu konzipiert von Maschinen gelesen zu werden
- Soll ebenfalls zum Datenaustausch zwischen wissenschaftlicher und mathematischer Software dienen

Anwender

- Lehrkräfte
- Studenten
- Ingenieure
- Wissenschaftler

Jetzt steht z.B. Professoren die Möglichkeit zur Verfügung, ihr wissenschaftlichen Arbeiten samt mathematischer Formeln in einem offenen und generell zugänglichen Web-Format anzubieten.

XML-Elemente und -Attribute (werden gemeinsam als Markup bezeichnet) lassen sich in zwei Kategorien einteilen:

Darstellungs-Markup	**Inhalts-Markup**
(presentation Markup)	(Content Markup)
beschreibt mathematische Ausdrücke von ihrem zweidimensionalen Layout	soll die Bedeutung des mathematischen Ausdrucks beschreiben

Da der Umfang des mathematischen Wissens und mathematischer Bedeutungen ständig wächst, ist es unmöglich, die Bedeutung aller mathematischer Daten durch 50 MathML-Elemente und ihre Attribute zu beschreiben. Um den Umfang des Inhalts-Markups auf eine handhabbare Größe zu beschränken, haben die Entwickler von MathML die mathematischen Daten, die beschrieben werden sollen, auf ungefähr Gymnasialebene beschränkt. Dadurch lassen sich wahrscheinlich die meisten mathematischen Informationen ausdrücken, die Sinnvollerweise zwischen Computerprogrammen ausgetauscht werden, die mathematische Formeln erstellen und/oder empfangen und damit Berechnungen durchführen.

Für Bereiche, in denen es nicht relevant oder praktikabel ist, mathematische Bedeutungen auszudrücken, enthält MathML den Darstellungs-Markup. Viele mathematische Ausdrücke sind mehrdeutig, wenn sie nicht von Menschen als Autoren und Lesern interpretiert werden. Selbst das funktioniert nur, wenn ein Bezug zu einem Teilgebiet von Mathematik oder Wissenschaft hergestellt wird. So kann beispielsweise ein Balken über einem Buchstaben in der Elektronik den Kehrwert eines Signals bedeuten, in anderen Bereichen den Wert der Variablen im letzten Schritt eines iterativen Algorithmus. Darstellungs-Markup dient lediglich dazu, mathematische Ausdrücke so zu beschreiben, dass sie von einem Webbrowser (oder einem Zusatzmodul eines Webbrowsers) angezeigt werden können. Die Darstellungs-Tags beginnen in der Regel mit "m" und verwenden ein "o" für den Operator, ein "i" für die Kennung, "n" für Zahlen usw. Die mrow-Tags dienen der Einteilung in horizontale Gruppen.

3.3 Die Vor- und Nachteile von MathMl

Vorteile:

- einfach und schnelles erstellen
- Formeln können zeitsparend in ein HTML Dokument eingefügt und wieder geändert werden
- Für alle anwendbar die Grundkenntnisse in HTML haben
- Kleinere Dateigröße gegenüber Bildern und daher höhere Ladegeschwindigkeit
- Keine Qualitätsverluste im Ausdruck, da kein Pixelinformation

Nachteile:

- Grundkenntnisse von HTML / MathML notwendig
- für manuelle Editierung zu „unhandlich“
- durch die Syntax nur schwer bzw. gar nicht durch CSS formatieren
- Standardisierung MathML noch nicht vollendet
- Nicht alle Browsertypen unterstützen MathML
- Es entsteht ein wahnsinnig aufwändiger Quelltext für nur eine Formel
    - Lösung: Editoren

3.4 Was MathML verspricht

Wenn MathML sich stärker etabliert hat (d. h., MathML wird von allen Browsern unterstützt, und es wird in Rechenanwendungen unterstützt), lassen sich damit hervorragende Anwendungen erstellen:

Fachdokumente

Durch richtige Browserunterstützung für MathML wird es möglich, Fachdokumente wie Zeitschriftenartikel als Webseiten zu erstellen. Diese sind besser als solche, die mit zurzeit verwendeten Methoden wie PDF (Adobe Acrobat), IBM Techexplorer, Mathematica Reader usw. erstellt werden, da diese Programme das gesamte Browserfenster belegen. Mit MathML können Anwender mathematische Daten aus Webseiten in ihre eigene Arbeit kopieren und für weiterführende Berechnungen und Analysen verwenden.

Praktisch angewendete Mathematik

Es wird möglich sein, Webseiten mit virtuellen Taschenrechnern zu erstellen, die mathematische Ausdrücke im mathematischen Standardformat ausgeben. Andere Seiten können mathematische, wissenschaftliche und technische Begriffe darstellen, wobei der Anwender Zahlen und mathematische Ausdrücke eingibt und nach dem Klicken auf eine Schaltfläche die Ergebnisse in einer grafischen Darstellung erhält.

Unterrichtshilfsmittel

Lehrkräfte können Prüfungen als Webseiten erstellen und sich dabei die oben beschriebenen Techniken zunutze machen. Damit niemand schummelt, können Zahlen und Variablen algorithmisch geändert werden, so dass jeder Prüfungskandidat eine etwas andere Prüfung erhält, wobei sich weiterhin richtige von falschen Antworten unterscheiden lassen.

3.5 Browsereinstellungen – was wird benötigt?

Um Seiten in MathML anzeigen zu können benötigen sie einen Browser der von Haus aus oder durch zu Hilfenahme eines Plug-ins MathML anzeigen kann.

Browserkonfigurationen für die vollwertige Ansicht von MathML
(ein + signalisiert das auch spätere Versionen funktionieren)

- Windows:
  - IE 5.0 mit dem Techexplorer Plug-in
  - IE 5.5 mit entweder den MathPlayer oder Techexplorer Plug-ins
  - IE 6.0+, optional mit MathPlayer oder Techexplorer Plug-ins
  - Netscape 6.1 mit Techexplorer Plug-in
  - Netscape 7.0+
  - Amaya, alle Versionen (nur Presentation MathML)
  - Mozilla 0.9.9+
- Macintosh:
  - IE 5.0+ mit dem Techexplorer Plug-in
  - Mozilla 0.9.9+
- Linux/Unix:
  - Netscape 6.1 mit dem Techexplorer Plug-in
  - Netscape 7.0+
  - Mozilla 0.9.9+
  - Amaya, alle Versionen (nur Presentation MathML)

Momentan am besten geeigent: Mozilla Firefox

- am weitesten entwickelt

- plattformunabhängig und ohne Plug-in

Amaya

Eventuell werden noch Schriftarten und Zeichensätze benötigt. Es erscheint eine Fehlermeldung, fall die Schriftarten noch installiert werden müssen. Die Schriftarten und Installationsanleitung finden sie bei Mozilla.

3.6 Einschränkungen für aktuelle Webbrowser-Zusatzprogramme

Es sind zwar Browser-Zusatzprogramme verfügbar, die MathML darstellen können, jedoch bieten sie keine völlig zufrieden stellende Lösung für die Darstellung von MathML auf Webseiten. Für alle gelten eine oder mehrere der folgenden Einschränkungen:

- Begrenzter Zeichensatz. Viele mathematische Symbole können nicht dargestellt werden.
- Die Darstellung von Zeichen ist davon abhängig, dass bestimmte Schriftarten auf dem Client-System installiert sind.
- Das Drucken ist auf die Bildschirmauflösung begrenzt, also ist das Ergebnis nicht besser als bei GIF-Bildern.
- Enthält eine Textzeile mathematische Daten, werden diese nicht sauber an der Grundlinie des Textes ausgerichtet.
- Jedes Zusatzprogramm erfordert "Verbindungscode" um jede Formel, um das Zusatzprogramm zu aktivieren.

- Bei der Darstellung von Formeln werden Eigenschaften der Browserumgebung wie die vom Anwender vorgenommene Schriftartauswahl, Bildschirm- und Dokumentbreite usw. nicht berücksichtigt.
- Schwerfällige Dokumentvorbereitung.
- Keine Links zu Berechnungs-Engines wie Maple und Mathematica.

3.7 MathML Software Editors

- Amaya
- EdiTex
- EzMath
- Formulator
- Integre MathMl Equation Editor
- Jscl-meditor
- MathCast
- MathFlow Editor for Arbortext
- MathFlow Editor for XMetal
- Mathiwyg Flash MathMl Editor
- Mathmled – MathMl Editor for Mozilla
- MathMl StudioLite
- MathType
- OpenOffice
- Publicon
- sMArTH
- Scientific Word
- Sci Writer
- WebEQ
  xfy technology
- WéM - PHP
- WéM - Web éditeur Mathématique

In dieser Ausarbeitung werden wir näher auf den Editor MathType eingehen.

3.8 Erstellen von MathML Formeln:

Eine Formel kann in zwei unterschiedlichen Modi notiert werden:

Präsentationsmodus:
- die Darstellung einer Formel wird festgelegt
- eine Formel besteht lediglich aus einer Folge von Symbolen, die in einem zweidimensionalem Schema angeordnet sind
- Symbole können neben-, unter-, übereinander positioniert, höher oder tiefergestellt werden

Beispiel: So wird x^2 einfach als x mit einer höhergestellten 2 notiert

```
<msup>
 <mi>x</mi>
 <mi>2</mi>
</msup>
```

semantischer Modus:
- die Bedeutung einer Formel wird notiert
- für die Einbindung in Xhtml ist der Präsentationsmodus zwar ausreichend, die im semantischen Modus notierten Formeln können jedoch auch von Computeralgebra Programmen interpretiert und gegebenenfalls ausgewertet werden

Beispiel: x^2 würde also je nach Bedeutung unterschiedlich notiert

```
<apply>>
 <power/>
   <ci type="real">x</ci>
   <ci type="integer">i</ci>
</apply>

<ci>
  <msup>
    <mi>x</mi>
    <mi>i</mi>
  </msup>
</ci>
```

Da über die Darstellung der im **semantischen Modus** notierten Formeln die jeweiligen Programme bzw. Browser entscheiden, gibt es noch den gemischten Modus, in dem sowohl ihre Bedeutung, als auch die genaue Darstellung angegeben werden, was sich vor allem bei Verwendung von eigen(willig)er Notation als sinnvoll erweist.

Überall wo in Xhtml normaler Text stehen kann, ist es möglich, auch Formeln einzufügen (also auch in Überschriften, Tabellen, Verweisen etc.). Hierzu muß lediglich der Code für die Formel innerhalb des `<math>` -Tags stehen:

```
<h1>
 Überschrift, die das Integral
 <math xmlns="http://www.w3.org/
1998/Math/MathML">
  <mrow>
   <msubsup>
    <mo> &int; </mo>
    <mn> 0 </mn>
    <mn> 1 </mn>
   </msubsup>
   <mrow>
    <msqrt><mi> x </mi></msqrt>
    <mo> &InvisibleTimes; </mo>
    <mrow>
     <mo> &DifferentialD; </mo>
     <mi> x </mi>
    </mrow>
   </mrow>
  </mrow>
 </math>
 enthält
</h1>
```

Wichtig ist, daß das **<math>**-Tag das Attribut xmlns="http://www.w3.org/1998/Math/MathML" enthält. Es handelt sich dabei nicht um einen Verweis auf eine Internetseite, sondern legt einen neuen sogenannten Namensraum fest: alle innerhalb von **$$** notierten Elemente werden als MathML- und nicht als Xhtml-Sprachelemente interpretiert.

Um Zugriff auf die benannten Symbole (∫ ⅆ) zu erhalten, muss auch die entsprechende Dtd-Datei (Dokumenttypdeklarationen) eingebunden werden; eine Stylesheet-Datei steuert die Ausgabe im semantischen Modus - und schon steht der vernünftigen Darstellung zumindest mit Mozilla 1.0 nichts mehr im Weg.

Zur Erstellung von MathML Code benötigen Sie einen MathML fähigen Formeleditor wie z.B. MathType. MathType ist ein komfortabler und leicht zu bedienender Formeleditor mit graphi-

scher Benutzeroberfläche. Die Formeln können mit MathType beim Kopieren in die Zwischenablage in MathML konvertiert werden, und dann direkt in das Formularfeld MathML-Code eingefügt werden.

3.8.1 Beispiele:

```
<mfenced>
  <mrow>
    <mi>a</mi>
    <mo>+</mo>
    <mi>b</mi>
  </mrow>
</mfenced>
<msup>
  <mn>2</mn>
</msup>
```

Das Beispiel bewirkt die Ausgabe: $(a+b)^2$

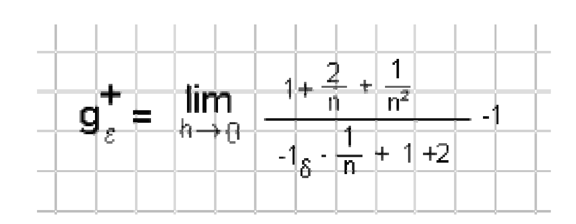
Der Aufwand dafür mag im ersten Augenblick übertrieben anmuten. Der Vorteil der Beschreibung in MathML ist jedoch, dass jedes einzelne Element des Ausdrucks eine eigene logische Auszeichnung erhält. Nur dadurch ist es möglich, auch beliebig komplexe Ausdrücke zu beschreiben.

Tag		Erläuterung
<mfenced>...</mfenced>	à	definiert die Klammer
<mrow>...</mrow>	à	definiert einen horizontal notierten zusammenhängenden Ausdruck (das was hier im Quelltext untereinander steht erscheint später nebeneinander)
<mi>...</mi>	à	definiert einen Bezeichner (*Identifier*) (in diesem Fall a und b)
<mo>...</mo>	à	definiert einen Operator (ein diesem Fall +)
<mn>...</mn>	à	definiert einen numerischen Wert (in diesem Fall die Zahl 2)
<msup>...</msup>	à	definiert diese Zahl als hochgestellt (Superscript)

<u>Ein etwas komplexeres Beispiel:</u>

```
<math>
  <mrow>
    <msubsup>
      <mtext>g</mtext>
      <mtext>n</mtext>
      <mtext>+</mtext>
    </msubsup>
    <mtext> = </mtext>
    <munder>
      <mtext>lim</mtext>
      <mtext>h&RightArrow;&Superset;0</mtext>
    </munder>
    <mtext> </mtext>
    <mfrac>
      <mrow>
        <mtext>1+</mtext>
        <mfrac>
          <mtext>2</mtext>
          <mtext>n</mtext>
        </mfrac>
        <mtext>+</mtext>
        <mfrac>
          <mtext>1</mtext>
          <msup>
            <mtext>n</mtext>
            <mtext>2</mtext>
          </msup>
        </mfrac>
      </mrow>
      <mrow>
        <msub>
          <mtext>-1</mtext>
          <mtext>δ</mtext>
        </msub>
        <mtext>-.</mtext>
        <mfrac>
          <mtext>1</mtext>
          <mtext>n</mtext>
        </mfrac>
        <mtext>+1+2</mtext>
      </mrow>
    </mfrac>
    <mtext>-1</mtext>
  </mrow>
</math>
```

Bewirkt die Ausgabe:



```
<mrow>
   <mi>x</mi>
   <mo>=</mo>
   <mfrac>
      <mrow>
         <mrow>
            <mo>-</mo>
            <mi>b</mi>
         </mrow>
         <mo>&PlusMinus;</mo>
         <msqrt>
            <mrow>
```

```
               <msup>
                  <mi>b</mi>
                  <mn>2</mn>
               </msup>
               <mo>-</mo>
               <mrow>
                  <mn>4</mn>
                  <mo>&InvisibleTimes;</mo>
                  <mi>a</mi>
                  <mo>&InvisibleTimes;</mo>
                  <mi>c</mi>
               </mrow>
            </mrow>
         </msqrt>
      </mrow>
      <mrow>
         <mn>2</mn>
         <mo>&InvisibleTimes;</mo>
         <mi>a</mi>
      </mrow>
   </mfrac>
</mrow>
```

bewirkt die Ausgabe:

"Mitternachtsformel"x $= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Dieses sinnfreie Beispiel demonstriert die Integration von MathML in Xhtml. Das Bild zeigt die Darstellung der Datei in Mozilla 1.0 (mit vergrößerter Schrift).



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet
  type="text/xsl"
  href="http://www.w3.org/Math/XSL/mathml.xsl"?>
<!DOCTYPE html
 PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
 "http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
 <title>MathML Beispiel</title>
</head>

<body>
<h1>
 Ein Beispiel
</h1>
<p>Ein wenig Text mit einer kleinen Grafik:</p>
<div><img src="logo.png" /></div>
<p>Und ein
<a href="http://www.fcb.de">Link</a>
sollte auch nicht fehlen, denn jetzt kommt der Clou:

 <math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
   <msubsup>
    <mo>&int;</mo>
    <mn>0</mn>
    <mn>1</mn>
   </msubsup>
   <mrow>
    <mo>&weierp;</mo>
    <mo>&ApplyFunction;</mo>
    <mrow>
     <mo>(</mo>
     <mi>x</mi>
     <mo>)</mo>
    </mrow>
    <mo>&InvisibleTimes;</mo>
    <mo>&DifferentialD;</mo>
    <mi>x</mi>
   </mrow>
  </mrow>
 </math>

 eine Formel mitten im Text,
 als hätte sie schon immer dazugehört.
</p>

<p>Weil's so schön war,
 notieren wir auch die Bedeutung der Formel:

 <math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
   <int/>
    <bvar><ci>x</ci></bvar>
    <interval><cn>0</cn><cn>1</cn></interval>
    <apply>
     <co>&weierp;</co>
     <ci>x</ci>
    </apply>
  </apply>
 </math>

</p>
</body>
</html>
```

3.9 *Einbindung von MathML in Webseiten*

Um die Anzahl der mit MathML kompatiblen Plattformen zu erhöhen, sollte ein Dokument unter Benutzung der folgenden Regeln geschrieben werden. Beachten Sie, dass wenn Sie einen den WC3 Standards befolgenden Editor wie Amaya verwenden, dies alles automatisch gemacht wird.

1. Erstelle die Seite unter Verwendung von XHTML mit inline MathML

Die Webseite sollte mit XHTML und einem inline MathML Bereich geschrieben werden, wie das folgende Beispiel zeigt:

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body>
    <h1>Beispiel</h1>
    ....
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi><mo>+</mo><mn>3</mn>
    </math>
  </body>
</html>
```

Der MathML Bereich sollte sich nicht in einem <object> Tag, oder in einer separaten, mit dem <embed> Tag verlinkten Datei befinden.

2. Füge einen Stylesheet verarbeitenden Befehl hinzu

Die folgende Zeile (fett geschrieben) sollte an den Anfang der XHTML Seite eingefügt werden, vor der Öffnung des <html> Tag, aber nach der XML Angabe <?xml...?>, wenn diese existiert:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="http://www.w3.org/Math/XSL/mathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
...
```

Dies wird in den meisten Fällen funktionieren (*aber nicht mit dem Internet Explorer: aus Gründen der Sicherheit führt der IE keine XSLT Stylesheets aus, die nicht auf dem selben Server wie das XHTML+MathML Dokument gespeichert sind. Befolgen Sie in diesem Fall die 'off-line' Instruktionen unten. Lesen Sie auch das Ende des Dokuments für mehr Informationen zur Sicherheit des Internet Explorers*). Wie auch immer, es gibt Alternativen:

- **Off-line**: Wenn das Dokument, dass die mathematischen Formeln beinhaltet, für die lokale Betrachtung ohne Internetverbindung gedacht ist, sollte die Angabe sein:

  <?xml-stylesheet type="text/xsl" href="mathml.xsl"?>

  Und eine Kopie der folgenden Dateien sollten sich auf dem lokalen Computer bei dem Hauptdokument befinden: mathml.xsl, pmathml.xsl, ctop.xsl und pmathmlcss.xsl.

- **Presentation MathML**: wenn das Dokument nur Presentation MathML beinhaltet, werden nur die Dateien pmathml.xsl und pmathmlcss.xsl benötigt. Die Anzeige wird beschleunigt, wenn nur diese beiden Dateien verwendet werden.

3.9.1 Bestimmung der Voreinstellungen

Weil es möglich ist, dass einem Browser mehr als eine Methode zum übertragen von MathML Bereichen zur Verfügung steht (zum Beispiel, der IE mit dem Techexplorer Plug-in kann auch Ausdrücke unter Verwendung von „CSS Rendering“ darstellen), gibt es einem Weg, um die bevorzugte Methode anzugeben. Dies wird erreicht durch die Angabe eines Attributs genannt renderer, welches zu einem bestimmten Namensraum gehört. Zum Beispiel:

```
<?xml-stylesheet type="text/xsl" href="pmathml.xsl"?>

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:pref="http://www.w3.org/2002/Math/preference"
      pref:renderer="css">
<head>...</head>
<body>...</body>
</html>
```

spezifiziert, dass die CSS Rendering Methode für diese Seite bevorzugt wird. Werte für die renderer Attribute sind:

- css: überträgt den Ausdruck durch die Benutzung von CSS (kein Plug-in benötigt)
- mathplayer-dl: biete an, MathPlayer zu installieren, falls nötig.
- mathplayer: verwende das MathPlayer Verhalten.
- techexplorer-plugin: benutze das Techexplorer Plug-in.
- techexplorer: das Techexplorer Rendering wird bevorzugt.

Microsoft Internet Explorer Sicherheit

Unter den Standardeinstellungen erlaubt Internet Explorer keine XSLT Stylesheets für ein Dokument, wenn diese nicht auf dem gleichen Server wie das Dokument liegen. Deswegen wird

```
<?xml-stylesheet type="text/xsl" href="http://www.w3.org/Math/XSL/mathml.xsl"?>
```

nicht funktionieren. Stattdessen ist es notwendig, die Stylesheets auf dem selben Server wie das Dokument zu kopieren (wie oben beschrieben). Alternativ ist es möglich, die Sicherheitseinstellungen im Internet Explorer zu verändern. Dies können wir aber nicht empfehlen, weil es das komplette Surfen, nicht nur die Verwendung dieses Stylesheets beeinflusst.

3.10 MathType

à ist ein Leistungsstarkes interaktives Zusatzprogramm für Windows und Macintosh, mit dem man mathematische Ausdrücke in Textverarbeitungs- und DTP-Dukumenten, in Webseiten, in Präsentationen sowie in Tex-, LaTex- und MathML-Dokumenten erzeugt.

3.10.1 Wie funktioniert MathType:

Ganz einfach per Mausklick lassen sich komplexe mathematische Formeln erstellen und in ein beliebiges Dokument einfügen. Sie müssen keine Codewörter im Kopf haben. Die Symbole und Vorlagen für Ihre Formel wählen Sie einfach mit der Maus oder über die Tastatur aus.

Bereits während der Eingabe wendet MathType die Regeln für den mathematischen Schriftsatz an. Automatisch werden beim Formelaufbau Schriftart, Größe und Abstände gewählt und richtig platziert.

Sie können selbst bestimmen, welche Abstände MathType verwenden soll, oder mit einem Tastendruck zwischen automatischer Formatierung und Fließtexteingabe wechseln. Und wenn es ganz perfekt sein soll: In MathType lassen sich einzelne Formelelemente in Schrittgrößen von 1/4 Punkt verschieben.

MathType ist mit über 500 mathematischen Symbolen und Vorlagen ausgestattet, wie z.B. Brüche, Wurzeln, Summen, Integrale, Produkte, Matrizen und verschiedene Klammerarten Außerdem können Sie Zeichen aus beliebigen auf Ihrem System installierten Schriftarten verwenden. MathType enthält auch den *Euclid Math-Zeichensatz,* mit dem Sie Ihren Dokumenten ein konsistentes und professionelles Erscheinungsbild im TeX/LaTeX Computer Modern-Stil verleihen.

MathType kann MathMl erzeugen, das zum Verfassen von Webseiten mit mathematischen Ausdrücken verwendet werden kann. Dies wird durch eine Konvertertechnologie erreicht. MathType enthält mehrere Definitionsdateien für die MathMl konvertierung, diese erstellen Math-Ml-Darstellungstags.

Erzeugen eines MathMl-Codes durch MathType

1. Öffnen Sie MathType
2. Wählen Sie den Konverter für MathMl aus
3. Erstellen Sie eine Formel mit MathType und kopieren Sie diese in die Zwischenablage
4. Bringen Sie das XHTML-Dokument in den Vordergrund und fügen Sie die eben kopierte Formel ein
5. ihr MathMl Code wird automatisch in das Dokument eingefügt

3.11 Vergleich MathType für Windows und Formel-Editor

Unterschiede zwischen MathType und dem Formel-Editor auf einen Blick. Hier werden auch die Vorteile von Mathtype klar verdeutlicht.

	Funktion	**MathType**	**Formel-Editor**
1	Anzahl mathematischer Symbole und Vorlagen	500+	275
2	Benutzerdefinierbare Formelsymbolleisten	Ja	Nein
3	Unterstützung von CMYK-, RGB- und Spotfarben	Ja	Nein
4	Automatische Anpassung von Abständen, Formaten und Größen aller Formeln in einem Microsoft Word-Dokument	Ja	Nein
5	Formelnummerierung und –verweise in MS Word	Automatisch	Manuell
6	Benutzerdefinierbare Tastaturkürzel	Ja	Nein
7	Output in TeX, LaTeX, AMS-TeX, AMS-LaTeX	Ja	Nein

8	MathML-Output (1.0 & 2.0)	Ja	Nein
9	Euclid-Zeichensatz	Ja	Nein
10	Speichern von Formeln als GIF-Dateien	Ja	Nein
11	Kantenglättung bei GIF-Formeln	Ja	Nein
12	Speichern von Formeln im EPS-Format	Ja	Nein
13	Formeleinstellungen in Einstellungsdateien speichern	Ja	Nein
14	Unterstützung internationaler Zeichen und Tastaturen	Ja	Ja
15	Unicode-basierte Zeichendefinition	Ja	Nein
16	Formatierungslineal mit Tabstopps	Ja	Nein
17	Definierbare Klammerausrichtung	Ja	Nein
18	Benutzerdefinierbare Erkennung von Funktionen (sin, cos etc.)	Ja	Nein
19	Größe von Symbolleisten und Arbeitsbereich definierbar	Ja	Nein
20	Programmierbare Konverterfunktion für benutzerdefiniertes SGML/XML	Ja	Nein
21	DLL-Schnittstelle für bestimmte Funktionen	Ja	Nein
22	Hierarchische Ansicht der Formelstruktur	Ja	Nein
23	Online-Hilfe	Ausführlich	Ja
24	Benutzerhandbuch	Ja	Nein
25	Technischer Support	Ja	???

Erläuterungen zur Tabelle

Farbe

Gestalten Sie Formeln oder Formelteile farbig - besonders nützlich für Präsentationen (wie MS PowerPoint) und Webseiten. MathType unterstützt CMYK-, RGB- und Spotfarben für DTP-Software wie QuarkXPress, Adobe InDesign, PageMaker usw.

Benutzerdefinierbare Tastaturkürzel

Benutzerdefinierbare Tastaturkürzel für alle Symbole, Vorlagen und Befehle. Ein Tastaturkürzel ist eine Taste oder eine Kombination aus mehreren Tasten, bei der meist die Strg-, Umschalt- und/oder Alt-Taste gedrückt wird.

Benutzerdefinierbare Formelsymbolleiste

- Enthält Hunderte besonders häufig verwendeter Symbole, Ausdrücke und Formeln
- Von der Formel mit Drag & Drop in die Symbolleiste kopieren

- Symbolen, Vorlagen oder Ausdrücken Tastaturkürzel zuweisen
- Ausdrücke und Symbole mit Hilfe von benannten Registerkarten (z. B. "Geometrie", "Naturwissenschaft" usw.) ordnen
- Angedockte oder frei bewegliche Symbolleiste
- Drei Ansichtsgrößen zur Anpassung an Monitorgröße, Auflösung und Benutzereinstellung

Integrierte Konverter für TeX und LaTeX

MathType ist mit Konvertern ausgestattet, die Formeln in verschiedene TeX-Versionen konvertieren:

- Plain TeX (wie in "The TeX Book" von Donald Knuth beschrieben)
- AMS-TeX (Plain TeX mit Erweiterungen der American Mathematical Society)
- LaTeX (Version 2.09 und höher)
- AMS-LaTeX (LaTeX mit AMS-Erweiterungen)

So verwenden Sie die Konverter:

6. Öffnen Sie MathType und einen Texteditor oder Ihr TeX/LaTeX-System.
7. Wählen Sie einen Konverter (TeX, LaTeX, AMS-TeX, AMS-LaTeX) aus.
8. Erstellen Sie eine Formel mit MathType und kopieren Sie diese in die Zwischenablage.
9. Bringen Sie das TeX-Dokument in den Vordergrund und fügen Sie den TeX-Code ein.
10. Fahren Sie mit der Texteingabe im Dokument fort und wiederholen Sie die Schritte 3 und 4, wenn Sie eine weitere Formel einfügen möchten.

Sie müssen keine Programmiersprache beherrschen, keine Klammernpaarigkeit beachten und keine Stichwörter im Kopf behalten.

Benutzerdefinierbare Erkennung von Funktionen

Wenn Sie die Buchstabenfolge "sin", "cos", "lim" etc. tippen, erkennt MathType, dass es sich um Funktionen handelt, und wechselt automatisch in die Formatvorlage für Funktionen. Den von MathType erkannten Standardfunktionen können Sie weitere hinzufügen.

Lineal zur genauen Formatierung

Wie in Ihrer Textverarbeitung haben Sie auch in MathType ein Lineal zur genauen Formatierung zur Verfügung. Ob Sie Spalten an Operatoren oder Dezimalzeichen ausrichten oder Tabstopps zur leichteren Formatierung setzen - Ihre Dokumente werden ein professionelleres Erscheinungsbild bekommen. Eine vertikale Linie erscheint im Formelfenster, wenn Sie einen Tabstopp setzen oder bewegen. Dies erleichtert die Ausrichtung einzelner Formelelemente.

Unterstützung internationaler Zeichen und Tastaturen

- Mit dem in den asiatischen Versionen von Microsoft Windows integrierten IME-Editor (Input Method Editor) lassen sich asiatische Zeichen eingeben.
- Unterstützung lateinischer Zeichensätze
- Wenn mehrere "Tastaturen" installiert sind, können Sie MathType dazu veranlassen, beim Wechsel von Formatvorlagen automatisch das Tastatursystem zu wechseln.

Umfangreiche Unicode-basierte Zeichendatenbank

- Unicode (http://www.unicode.org) ist ein Zeichencodierstandard, der annähernd sämtliche in den existierenden Schriftsprachen und in der Mathematik verwendeten Zeichen enthält.
- Die MathType-Datenbank enthält Zeicheninformationen zu den meisten bekannten mathematischen Zeichensätzen.
- Beschreibungen der Zeichen werden in der Statusleiste angezeigt und können durchsucht werden, um das gewünschte Zeichen zu finden.
- Im Dialogfeld "Symbol einfügen" durchsuchen Sie die Schriftarten auf Ihrem Computer und zeigen Zeicheninformationen an.

Speichern von Formeln in den Formaten GIF, EPS (Encapsulated PostScript) und WMF (Windows Metafile)

GIF-Bilder

MathType erstellt sauber aussehende GIF-Formeln auf farbigem oder transparentem Hintergrund. Aufgrund der Kantenglättungsfunktion (Anti-Aliasing) sind in MathType erstellte mathematische Formeln von wesentlich besserer Qualität als Formeln, die Sie aus Microsoft Word heraus in HTML konvertieren. Auch die Downloadzeit für MathType-GIF-Dateien ist i. d. R. kürzer.

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

EPS und WMF

Sollen professionelle Druck-Erzeugnisse mit QuarkXpress, Adobe InDesign, Adobe PageMaker oder ähnlichen Seitenlayoutprogrammen erstellt werden, ist EPS meist das bevorzugte Format. Für Strichzeichnungen auf Windows-Systemen ist WMF das Standarddateiformat. MathType lässt die Speicherung von Formeln in beiden Formaten zu.

3.12 Besondere Funktionen von MathType

- MathType ist ausgestattet mit der neuen *MathPage™*-Technologie, die Ihre Microsoft Word-Dokumente spielend leicht in Webseiten konvertiert. Sie sorgt dafür, dass sowohl mathematische Symbole als auch in MathType *und* dem Formeleditor erstellte Formeln richtig gehandhabt und in einem modernen Windows-, Macintosh-, Unix- oder Linux-Browser korrekt dargestellt werden. Ihr gesamtes Word-Dokument wird in eine Webseite konvertiert; darin enthaltene Formeln werden im GIF- oder MathML-Format abgespeichert, und die Darstellungsqualität im Browser ist beeindruckend. Das Resultat ist weitaus besser als bei Adobe PDF oder der Word-Funktion *Als Webseite speichern*.

Vorteile dieser Funktion

- Die Seiten haben HTML-Standardformat mit JavaScript und lassen sich plattformunabhängig (Windows, Mac, Linux, Unix) mit jedem modernen Browser (Internet Explorer, Netscape, Mozilla) anzeigen.
- Keine Installation einer zusätzlichen Software auf dem Web-Server erforderlich. Es wird lediglich ein Browser benötigt.
- Auf Adobe Acrobat können Sie verzichten - Sie brauchen nur MathType und Microsoft Word.

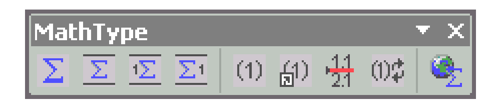
- Auf dem Rechner des Web-Besuchers müssen keine Plugins oder speziellen Mathematik-Schriften installiert werden (außer bei Generierung von MathML).
- Die Formel-Darstellung wird automatisch für die Bildschirmauflösung des jeweiligen Benutzers optimiert.
- MathPage verwandelt die Formeln in hochauflösende GIFs. Das Resultat ist eine exzellente Druckqualität anstelle plumper, grober Zeichen, bei denen Sie jeden Pixel erkennen.
- Besucher Ihrer Website können dort eingestellte Formeln weiterverwenden, indem sie diese via Drag & Drop oder über die Windows-Zwischenablage in ihre eigenen Dokumente einfügen.

Weitere Funktionen:

- Neuartige "verschachtelte" Ansicht zum besseren Erkennen der Formelstruktur
- Vergrößerung auf 800 % für noch genaueres Ausrichten
- Anzeige leerer Felder und Abstände in den gewünschten Farben
- Genauere Anzeige (Befehl zum Aktualisieren der Anzeige nicht mehr notwendig)
- Augenfreundlich einstellbare Größe der Symbolleiste
- Online-Versionscheck im Internet nach neuen Versionen von MathType
- Produktregistrierung online, via Internet-Browser oder E-Mail, oder wahlweise per Fax oder Post
- Direktzugriff auf die MathType-Webseite Technischer Support über das Hilfe-Menü
- Im Dialogfeld *Speichern unter* lassen sich Formeldateien ganz einfach automatisch durchnummerieren (z. B. "form1", "form2" usw.). Besonders praktisch ist dies bei der Arbeit mit GIF- oder EPS-Dateien.
- Das MathType-Benutzerhandbuch bietet jede Menge Tipps, Vorgehensweisen, Übungen und Beispiele. Sie sparen damit Zeit und kommen zu besseren Resultaten. In der Online-Hilfe finden Sie ausführliche Erläuterungen zu MathType-Befehlen und Dialogfeldern.

MathType-Symbolleiste

Diese Symbolleiste fügt MathType in Microsoft Word 97 (oder höher) ein. Damit können Sie Formeln abgesetzt oder im Fließtext platzieren, Kapitel, Abschnitte und Formeln nummerieren, die Nummerierung während der Bearbeitung aktualisieren, Formelverweise einfügen, Word-Dokumente in Webseiten verwandeln - und das alles direkt in Word.



MathType installiert außerdem folgende Befehle in Word:

- **Formeln formatieren** -- Automatische Anpassung von Abständen, Formaten und Größen aller Formeln in einem Microsoft Word-Dokument ohne Öffnen der einzelnen Formeldateien.
- **Formeln konvertieren** -- Konvertierung aller Formeln eines Dokuments in TeX, LaTeX oder MathML. Mit diesem Befehl konvertieren Sie auch im Formeleditor erstellte Ausdrücke in MathType-Formeln. Den Formeleditor brauchen Sie dann wirklich nie wieder.
- **Formeln exportieren** -- Automatischer Export aller Formeln in einen eigenen Ordner als EPS-, GIF- oder WMF-Dateien. Wenn Sie mit einer Seitenlayoutanwendung oder Software zur Erstellung von HTML-Seiten arbeiten, sparen Sie mit diesem Befehl viel Zeit, da Sie die Dateien nicht mehr einzeln öffnen und speichern müssen.

- **Formelnummern einfügen** -- Einfügen von Kapitel-, Abschnitts- und Formelnummern entweder rechts oder links der Formeln. Wenn Sie irgendwo im Dokument nachträglich eine neue Formel einfügen, wird die Formelnummerierung des gesamten Dokuments automatisch aktualisiert.
- **Formelnummern formatieren** -- Anpassen der Formelnummerierung: Römisches oder arabisches Nummernformat, Klein- und Großbuchstaben sowie beliebige, in der jeweils zugewiesenen Formatvorlage verfügbare Klammern und Trennzeichen.
- **Formelverweise einfügen** -- Durch Doppelklick auf einen Formelverweis springt die Einfügemarke auf die zugehörige Formel. Wenn Sie MathType zur Konvertierung eines Dokuments in eine Webseite verwenden, werden die Formelverweise automatisch zu Hyperlinks.
- **Einstellungsdateien** -- Sie können MathType-Einstellungen (Abstände, Formate, Größen) in Einstellungsdateien speichern und jederzeit in MathType oder Word damit arbeiten.

3.12.1 Konverterfunktion

Eines der wichtigsten neuen Merkmale ist die leistungsfähige Konvertierungsfunktion. Unter MathType wurde die Konvertierungsfunktion erweitert, so dass Formeln in viele andere Sprachen konvertiert werden können. Der Konvertierungsprozess wird von einer Konverterdefinitionsdatei gesteuert, einer Textdatei mit einfachen Konvertierungsbefehlen. Mit MathType kann eine beliebige Anzahl von Konverterdefinitionsdateien verwendet werden, was viele wichtige Vorteile mit sich bringt:

- Es können Konverter für praktisch alle Arten von mathematischer Beschreibung erstellt werden.
- Neue Konverter können von Grund auf neu oder durch das Umbenennen und Bearbeiten vorhandener Konverter erstellt werden.
- Anwender können Konverter per E-Mail und über das WWW untereinander austauschen.
- Design Science wird seine eigenen Konverter ständig aktualisieren und verbessern und sie über seine Website http://www.mathtype.com zur Verfügung stellen.

MathType enthält mehrere MathML-Konverter, und zwar einen für jedes der verfügbaren Webbrowser-Zusatzprogramme:

- **WebEQ-kompatibel:** WebEQ ist eine Suite von Java-Werkzeugen und -Applets zur Verarbeitung von MathML und zur Darstellung interaktiver mathematischer Inhalte im World Wide Web. WebEQ bietet nahezu vollständige MathML-Implementierung. Es fehlen nur wenige, besonders seltene Zeichen.
- **IBM Techexplorer-kompatibel:** Techexplorer ist ein Browser für wissenschaftliche Dokumente, der als Zusatz- oder Komplettprogramm installiert werden kann. Techexplorer bietet ausgezeichnete LaTeX-Unterstützung und ist sehr funktionsreich, bietet aktuell jedoch nur prototypische MathML-Unterstützung. Vor allem die Unterstützung von Symbolzeichen ist begrenzt. Einfache MathType-Formeln werden einwandfrei konvertiert und angezeigt, bei komplizierteren Ausdrücken jedoch stößt dieses Programm an seine Grenzen.
- **Amaya-kompatibel:** Amaya ist der Test-Browser des World Wide Web-Konsortiums W3C. Vorkompilierte Amaya-Binärdateien sind zwar nur für die Betriebssysteme Linux und Solaris verfügbar, Amaya ist jedoch mit vielen interessanten und hochaktuellen Funktionen ausgestattet, die das Resultat andauernder Web-Technologieentwicklung durch das W3C sind. Amaya bietet Unterstützung für die Darstellung von MathML, verwendet jedoch Objektnamen, die nicht dem Standard entsprechen.

3.12.2 Systemanforderungen

- Microsoft Windows 98, 2000, ME, und XP
- 10 MB freier Festplattenspeicher
- CD-ROM-Laufwerk zur Installation (nicht erforderlich bei Download via Internet)
- MathType ist nicht speicherintensiv. Wenn Sie Windows ausführen können, können Sie auch MathType ausführen.

4. LaTeX (Julia-Mareike Staff)

4.1 Was ist LaTeX?

LaTeX (gesprochen „LaTech“) ist ein Schriftsatzsystem, dessen Aufgaben und Möglichkeiten über die Erstellung mathematische Texte hinausgehen. Andere Einsatzgebiete sind das verfassen kurzer Mitteilungen, die private und geschäftliche Kommunikation, sowie die Erstellung wissenschaftlicher Arbeiten auf verschiedenen Gebieten. Versionen von LaTeX sind für alle Arten von Rechnern verfügbar.
Diesem Schriftsatzsystem liegt das Formatierungsprogramm TeX zugrunde, das Ende der 70er Jahre von Donald Knuth an der Stanford Universität entwickelt wurde. Zweck war es, ein Programm zu entwickeln, mit dessen Hilfe der Schriftsatz von Büchern, die eine große Anzahl von Formeln enthalten, grafisch ansprechend gestaltet werden kann. Eben aufgrund dieser Eigenschaft ist TeX (oder eben LaTeX) bei Wissenschaftlern weltweit sehr beliebt.
Neben dem Vorzug der Darstellung mathematischer Formeln, auf deren Darstellungsmöglichkeiten im Internet hier näher eingegangen werden soll, handelt es sich um ein portables System, das auf verschiedenen Rechnerplattformen lauffähig ist.

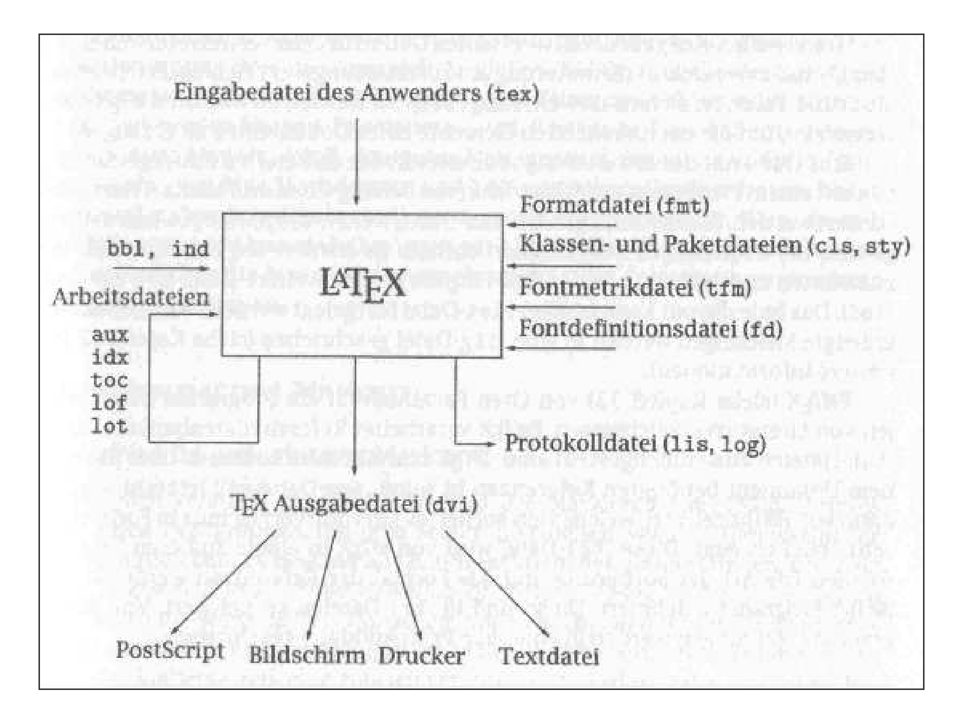
Zu Beginn der 80er Jahre wurde LaTeX auf Basis von TeX von Leslie Lamport entwickelt. Mit Hilfe von LaTeX können TeX-Basisbefehle abstrahiert werden, sodass sich der Anwender auf der Struktur des Dokuments konzentrieren kann, statt sich mit Formatierungsdetails zu befassen. Dies hat den Vorteil, dass weniger komplizierte Befehle zur Erstellung benötigt werden, da eine Anzahl an Layout-Vorlagen oder Styles zur Verfügung steht. Zudem ist LaTeX anders als TeX auch zur Erstellung von Indizes, Verzeichnissen und Verweisen in der Lage, wozu allerdings Hilfsprogramme nötig sind.

4.2 Wie funktioniert LaTeX?

Die Basis einer jeden LaTeX- Datei bildet die Eingabe- oder Quelldatei, die vom Anwender mit Hilfe eines Texteditors erstellt und mit der Endung „.tex“ gespeichert wird.

Darüber hinaus existieren Definitionsdateien für Struktur und Layout, die mit den Endungen „.sty“ und „.cls“ abgespeichert werden. Die Metrikangaben für das Dokument (d.h. der Zeichensatz der verwendeten Zeichen) werden auch Fontmetrik genannt und mit der Endung „.tfm“ erfasst und müssen für jeden verwendeten Zeichensatz vorhanden sein. Hier werden Höhe, Breite, Tiefe und Kerning- und Lignaturformen für den Satz festgelegt. Die Zuordnung interner Fontnamen erfolgt über die Fontdefinitionsdateien („.fd“).
Die Trennung von Wörten erfolgt in TeX automatisch nach einem Algorithmus, allerdings können für jede Sprache eigene Trennungsregeln festgelegt werden, wenn eine Formdatei („.fmt“) erzeugt wird (Enthält alle LaTeX-Befehle in kompilierter Form und .tfm-Dateien für einige Zeichensätze, die beim Start von LaTeX automatisch geladen werden.).
Die Ausgabe erfolgt über eine Reihe von Dateien. Eine dieser Ausgabedateien mit der Endung „.dvi“ enthält eine binäre Darstellung des formatierten Textes, in dem für jedes Zeichen Schriftart und Position definiert sind. Hier sind lediglich Fontnamen, keine formatierten Zeichen, enthalten. Um diese Datei grafisch darstellen zu können, wird ein sog. DVI-Treiber (z.B. PostScript) benötigt.
Bei jedem Lauf wird eine Protokolldatei erstellt, die die Endung „.log“ oder „.lis“ trägt, je nach Betriebssystem aber auch ohne Endung stehen kann. Diese Datei enthält z.B. die Namen der gelesenen Dateien, die Seitenzahl, Warnungen und Hinweise auf Fehler.
Weitere Ausgabedateien enthalten z.B. Daten für Querverweise („.aux“), Inhaltsverzeichnis („.toc“), Abbildungsverzeichnisse („.lof“) und Tafelverzeichnisse („.lot“). Sie werden bei jedem Lauf verwendet, um diese Elemente zu erzeugen.
Alle indizierten Einträge sind in der Datei „.idx“ enthalten, diese können z.B. mit dem Programm „MakeIndex“ sortiert werden. Literaturverzeichnisse können mit Hilfe von „BIBTeX“ erstellt werden und unter „.bib“ gespeichert werden.

Dieser Aufbau kann anhand folgender Abbildung dargestellt werden:



Um das so erzeugte Dokument letztendlich darstellen zu können, gibt es verschiedene Möglichkeiten:

1. die Datei mit einer Auflösung über 1000 dpi ausdrucken
   (Übersetzung der dvi- Datei in Punkte auf einem Film à Speichern als Bilddatei)
2. die Datei mit mittlerer Auflösung ausdrucken, dazu wird ein druckerspezifischer dvi- Treiber benötigt (z.B. PostScript)
3. den Inhalt mittels Programmen wie Xdvi (für XWindows) auf einem Grafikbildschirm darstellen (à PostScript- Previewer)
4. Die Darstellung ist auch auf einem Bildschirm möglich, der nicht Grafikfähig ist, wobei in diesem Fall keine Grafiken dargestellt werden können und der Text umformatiert wird.

Möchte man mathematische Formeln im Web darstellen, bietet es sich an, die LaTeX- Datei in ein MathML- Format umzuwandeln und in den Quellcode der Website einzubinden, wobei die Darstellung dann allerdings nur in einem MathML-fähigen Browser (z.B. Mozilla Firefox) erfolgen kann. Der Microsoft Internet Explorer ist dazu nicht in der Lage (bzw. nur mit Plugins).
Dargestellt werden sollen hier Möglichkeiten und Beispiele zur Erstellung von LaTeX-Dateien, Möglichkeiten zur Umwandlung in MathML und Gründe, warum eine Erstellung der Formeln in LaTeX vorteilhafter ist als die direkte Erstellung in MathML.

4.3 Wie wird eine LaTeX- Datei erzeugt?

Die Eingabe einer LaTeX- Datei kann über einen beliebeigen Texteditor erfolgen, die „Versuche“ zum erstellen dieser Arbeit wurden mit dem kostenlosen LaTeX- Editor „WinShell“ durchgeführt, welcher sich durch seine Übersichtlichkeit und die für Windows-Nutzer vertraut wirkende Oberfläche auszeichnet. Zudem ist das Programm in deutscher Sprache verfügbar, was Anfängern das Arbeiten mit Sicherheit zusätzlich erleichtert.
Weiter gibt es eine Unterstützung für BIBTex, eine Rechtschreibüberprüfung und zur Übersichtlichkeit Möglichkeiten für Hervorhebung im Quelltext. Weitere derartige Tools sind emacs, TeXnicCenter und NEdit.
Um grundsätzlich das Vorgehen zu erläutern soll zunächst einmal das Beispiel einer „einfachen“ Textdatei erfolgen:

```
 %&latex
\documentstyle[german]{article}
\begin{document}
\pagestyle{empty}
{\bf Hier findet Ihr Infos zum Thema Mathematische Formeln im Web.}
\end{document}
```

%&latex zeigt dem Rechner, dass es sich bei dieser Datei um ein LaTeX-Dokument handelt, und dass dieses auch so behandelt werden soll.
\documentstyle kennzeichnet durch [german]zum einen, dass es sich um ein deutschsprachiges Dokument handelt (wichtig z.B. wegen dem Vorhandensein von Umlauten (ä,ü,ö,…) usw. in der deutschen Sprache), aber auch, um welche Art von Dokument es sich handelt, hier zum Beispiel {article}, das sich gut eignet für kurze Texte. Weitere Möglichkeiten sind zum Beispiel {report} für mittellange Texte und {book} für lange Texte.
\begin{document} kennzeichnet den Anfang des eigentlichen Dokuments, \pagestyle{empty} legt den Stil der Seite fest, dies ist hier eine leere Seite. Mit plain erscheint eine leere Kopfzeile und in der Fußzeile die Seitenzahl.
In der nächsten Zeile steht der eigentliche Inhalt der Datei in { }. Die Formatierung \bf steht für boldface (fett). Die Ausgabe sähe dementsprechend in etwa so aus:

Hier findet Ihr Infos zum Thema Mathematische Formeln im Web.

Wird nichts weiteres angegeben, ist die Standardschriftart Times New Roman (roman). Andere Schrifttypen lassen sich festlegen durch Formatierungen wie \sf (sans serif), \sc(small caps, „Kapitälchen“) usw. Außerdem kann der Text kursiv (\it) gesetzt oder unterstrichen werden (\underline).
Weiter möglich sind die Buchstaben hinsichtlich ihrer Größe formatierbar. Durch \tiny werden diese winzig dargestellt, durch \scriptsize klein, durch \large groß usw.
Das Drücken der Enter-Taste erzeugt keinen Zeilenumbruch, d.h.:

```
Hallo
Welt
```

Erzeugt die gleiche Ausgabe wie `Hallo Welt`.
Normalerweise erfolgen die Zeilenumbrüche im fertigen Dokument automatisch. Möchte man diesen erzwingen sollte man entweder eine Leerzeile lassen oder aber einen Absatz setzen, d.h. an der Stelle, an der der Umbruch erfolgen soll, wird dies als Absatz gekennzeichnet mit \par.
Damit das Dokument dargestellt werden kann, muss es zwingend beendet werden. Dies geschieht mit: \end{document}

4.4 Mathematische Formeln in LaTeX

Nachdem das grundsätzliche Vorgehen beim erzeugen einer LaTeX-Datei anhand eines Textes dargestellt wurde, soll nun erläutert werden, wie mathematische Formeln in LaTeX erzeugt werden können. Hier gibt es eine unglaubliche Fülle an Möglichkeiten, wobei hier leider nur einige Beispiele dargestellt werden können, da es in erster Linie um die Darstellung dieser Formeln im Web gehen soll. Wessen Interesse an LaTeX durch diese Ausarbeitung geweckt wurde und wer gern selbst einmal damit auseinandersetzen möchte (für Diplomarbeiten o.ä.), dem sei an dieser Stelle die Seite http://www.uni-giessen.de/hrz/tex/cookbook/cookbook.html ans Herz gelegt. Hier findet sich eine umfangreiche, schrittweise Einführung in die Thematik, wobei am Ende eines jeden Kapitels die Möglichkeit zum Ausprobieren besteht. Hier werden auch die umfangreichen Möglichkeiten zur Darstellung mathematischer Formeln ausführlich dargestellt.

Grundsätzlich ist die Struktur eines Dokuments mit mathematischen Formeln in LaTeX dieselbe wie in der einfachen Textdatei. Das weitere Vorgehen zur Darstellung mathematischer Formeln im Web soll an folgendem Quelltext erläutert werden:

```
%&latex
\documentstyle[german]{article}
\begin{document}
\pagestyle{empty}
$\sqrt{a+b} = \sqrt[5]{c+d}$
\end{document}
```

Wie zu erkennen ist, unterscheidet sich der hier dargestellte Quelltext lediglich durch die Formel von der Textdatei. Die Ausgabe würde hier lauten:

$\sqrt{a+b} = \sqrt{[5]}c + d$

Die Formel ist in diesem Quelltext abgesetzt bzw. gekennzeichnet durch $...$. Grundsätzlich gibt es hierzu aber 2 Möglichkeiten. Und zwar:

```
$Hier die Formel$,
```

eben wie oben (Fließtext-Formel, z.B. in laufenden Text eingefügt), und

```
/[Hier die Formel/]
```

Die Absetzung durch Dollarzeichen eignet sich insbesondere für kurze, die Absetzung mit / [für längere Formeln.

Plant man, eine größere Anzahl an Formeln in einen Text einzubinden, bietet es sich an, die Formeln zu Nummerieren und Sie über die Nummerierung zu identifizieren. Dies lässt sich erzielen durch:

```
\begin{equation}
  Hier die Formel
\end{equation}
```

Fließtext-Formeln werden anders dargestellt als Formeln, die durch die Klammerumgebung gekennzeichnet werden. LaTeX versucht die vertikale Größe einer Fließtext-Formel so gering wie möglich zu halten. Das folgende Beispiel soll dies verdeutlichen:

Hier eine `\[Formel\]`:

$$\int_0^5 x$$

Und hier eine `$Formel$`:

$\int_0^5 x$

Die Größe von mathematischen Formeln werden von LaTeX automatisch ermittelt und die Formeln skaliert. Das Ergebnis dabei ist fast immer perfekt. Manchmal kann es allerdings notwendig sein, in die Skalierung einzugreifen. Dies ist z.B. bei Klammern manchmal notwendig:

$$t = \ln(\frac{U_C(0)-U_0}{U_0}) \cdot RC$$

$$t = \ln\left(\frac{U_C(0)-U_0}{U_0}\right) \cdot RC$$

Bei der oberen Formel wurde darauf verzichtet, die beiden Klammern an dem Bruch auszurichten. Die Skalierung der Klammern kann erreicht werden, indem man die Klammer mit \left(und \right) auszeichnet (wie bei der unteren Formel).

Variablen werden in Formeln grundsätzlich kursiv dargestellt. Da dies bei speziellen Abkürzungen von Funktionen (wie z.B. ln bei der obigen Formel) meist nicht erwünscht ist, können solche Abkürzungen mit einem Backslash von der Kursiv-Formatierung ausgenommen werden.

Funktionsnamen, die nicht kursiv dargestellt werden (Beispiele):

```
\arccos
\arctan
\cos
\deg
\dim
\gcd
\lim
\ln
\log
\max
\min
\sin
\tan
```

Für den Fall, dass Text in Formeln dargestellt werden soll, muss dieser mit

```
\mbox{Text in Formel}
```

gekennzeichnet werden. Da der Text ohne Zwischenräume in die Formel eingebunden wird, sollte man hier manuell die Zwischenräume mit

`\ \mbox{Text in Formel} \` festlegen.

$$U_C(t) = U_0 \cdot (1 - e^{-\frac{t}{\tau}}) mit \tau = R \cdot C$$

$$U_C(t) = U_0 \cdot (1 - e^{-\frac{t}{\tau}})\ \mbox{mit}\ \tau = R \cdot C$$

Bei der oberen Formel wurde das Wort "mit" nicht mit \mbox{mit} von der Kursivformatierung ausgenommen. Bei der unteren Formel wurde zusätzlich noch jeweils ein (zugegeben zu kleines) Leerzeichen vor und nach \mbox{mit} eingefügt.

Formeln werden in LaTeX durch Anweisungen konstruiert, so dass für eine Vielzahl von Formeln eigene Anweisungen existieren. Die Anweisungen für eine Formel werden meistens durch naheliegende Befehle realisiert. So wird z.B. eine Integral von 0 bis 5 der Funktion x durch folgenden Befehl umgesetzt:

```
/[ \int_{0}^{5} x /]
```

Selbstverständlich lassen sich Formeln in LaTeX beliebig schachteln. Statt der Funktion "x" hätte man also auch eine weiteres Intergral verwenden können.

Benannte Funktionen in LaTeX sind zum Beispiel:

`x^{y}` x^y

`x_{y}` x_y

Hierbei handelt es sich jedoch immer nur um einige wenige Beispiele, um kurz darzustellen, wie der Aufbau einer Formel grundsätzlich funktioniert.
Weiter gibt es Möglichkeiten, die Abstände in und zwischen Formeln festzulegen und mathematische Akzente zu setzen, sowie eine Vielzahl an Möglichkeiten bezüglich verschiedener Symbole, die aus Gründen der Überschaubarkeit hier nicht weiter erläutert werden sollen.

4.5 Wie werden mathematische Formeln in LaTeX webfähig?

Grundsätzlich ist LaTeX ein Format, das in einem Browser nicht dargestellt werden kann, also an sich in seiner „ursprünglichen" Form nicht webfähig ist. Um z.B. Formeln, die mittels LaTeX erzeugt wurden, in eine Webseite einzubinden, gibt es verschiedene Möglichkeiten.
Die einfachste ist mit Sicherheit, die jeweilige Formel umzuwandeln in eine Grafikdatei (z.B. .gif) und die so entstandene Grafik in das HTML-Dokument der Seite zu integrieren. Nachteil hierbei ist allerdings, dass diese Grafiken statisch sind, d.h. es ist beispielsweise nicht möglich, dass die Formel entsprechend Ihre Farbe verändert, wenn man die Maus darüber bewegt. Auch werden durch derartige Grafiken die Ladezeiten der Seite verlängert.
Die weitaus empfehlenswertere Variante ist allerdings, die LaTeX-Datei zu erzeugen und diese über verschiedene Konverter in MathML umwandeln zu lassen, was wiederum in den HTML-Code eingebunden wird. Die Möglichkeiten hierzu sollen im Anschluss erläutert werden.
Vorab ist allerdings die Frage zu klären, welche Vorteile es bringt, eine Formel in LaTeX zu erstellen und dann in MathML umzuwandeln anstatt sie direkt in MathML zu erstellen. Dies lässt sich am einfachsten plastisch darstellen, indem man sich die den einfachen, kurzen Quelltext zur Wurzelfunktion aus dem vorangegangenen Abschnitt zur Hand nimmt:

```
%&latex
\documentstyle[german]{article}
\begin{document}
\pagestyle{empty}
$\sqrt{a+b} = \sqrt[5]{c+d}$
\end{document}
```

Der Quelltext ist kurz und knapp und kann nach kurzer Einarbeitung in LaTeX einfach und schnell erstellt werden. Die gleiche Formel sähe in MathML etwa so aus:

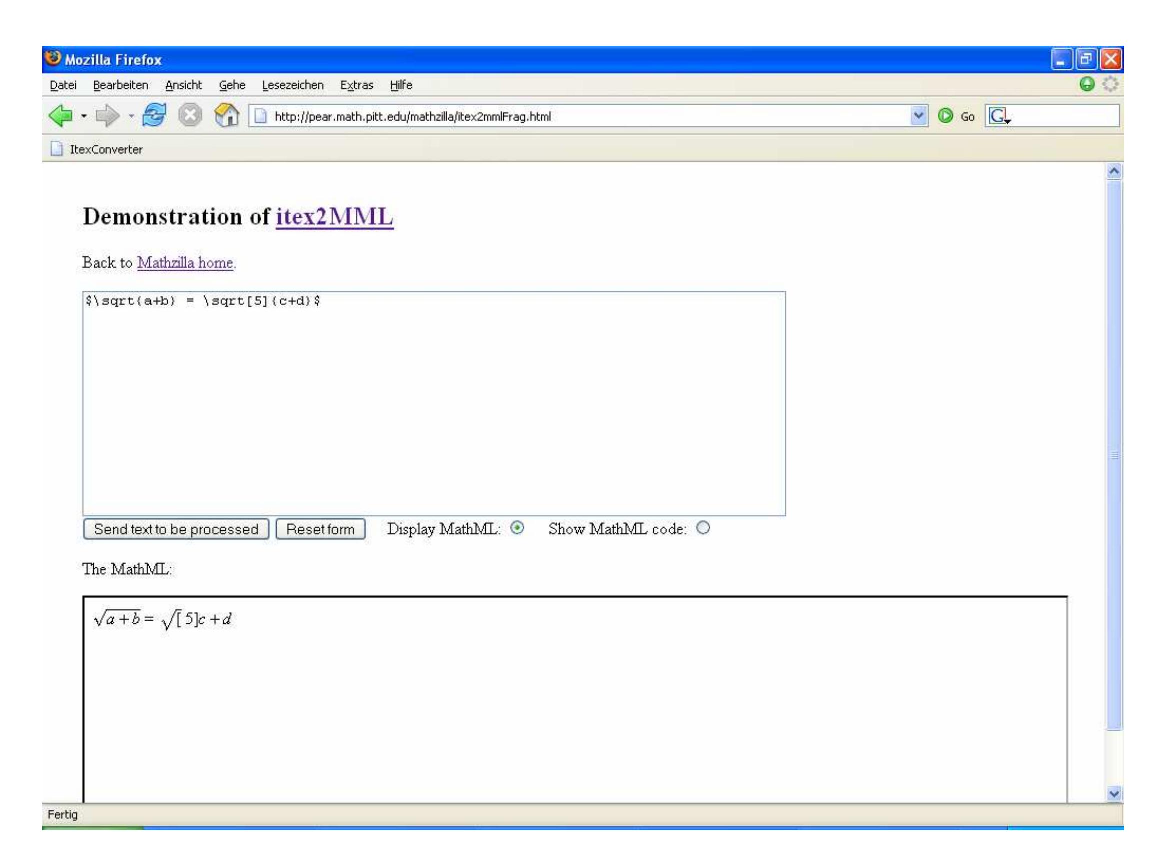
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
```

```
</head>
<body>
 <math xmlns='http://www.w3.org/1998/Math/MathML'>
  <msqrt>
   <mrow>
    <mi>a</mi>
    <mo>+</mo>
    <mi>b</mi>
   </mrow>
  </msqrt>
<mo>=</mo>
   <msqrt>
     <mo>[</mo>
   </msqrt>
       <mn>5</mn>
       <mo>]</mo>
   <mrow>
     <mi>c</mi>
     <mo>+</mo>
     <mi>d</mi>
   </mrow>
  </math>
</body>
</html>
```

Diese Datei ist in Ihrem Aufbau nicht nur komplexer, sie ist auch wesentlich länger. Was bei dieser kurzen Datei noch relativ wenig ins Gewicht fällt, kann bei komplexeren Formeln dazu führen, dass Formeln, die in LaTeX mit 2-3 Zeilen formatiert werden können, in MathML mehrere Seiten umfassen, die in sich zusätzlich stark verschachtelt sein können.

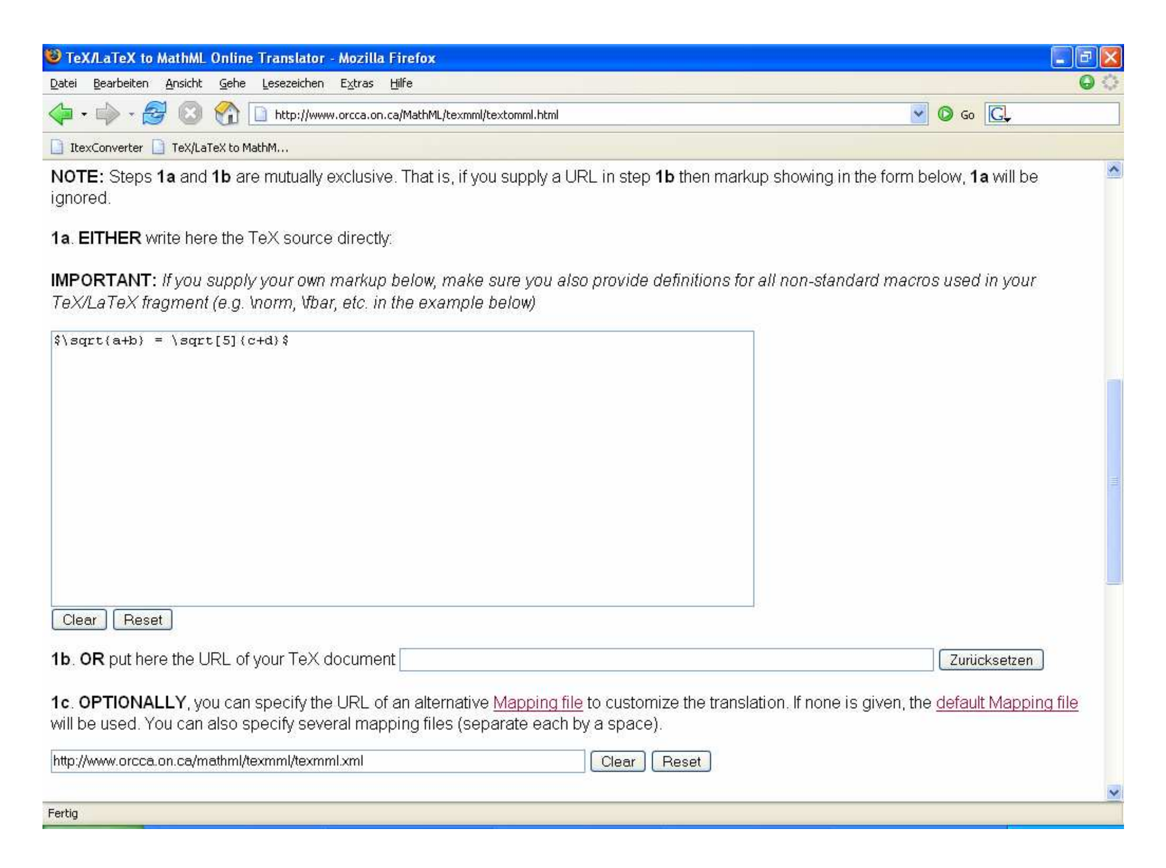
Um eine LaTeX- Datei in MathML umwandeln zu können, ist es erforderlich, die Datei zunächst auf dem lokalen Rechner zu speichern (Sicherheitshalber!).
Für die anschließende Umwandlung gibt es nun verschiedene Möglichkeiten. Zu einen kann man die Dateien an verschiedenen Stellen online konvertieren lassen, zum anderen gibt es spezielle Programme, um Dateien offline zu konvertieren, welche auf dem Rechner installiert werden müssen. Hier sollen jedoch lediglich die Online-Tools beschrieben werden, da die Offline- Programme relativ komplex sind und zum Teil nur mit erheblichem Aufwand installierbar und ausführbar (im Rahmen des Versuches z.T. auch nicht lauffähig). Hier sind die Online- Konverter wesentlich einfacher und komfortabler.
Beispiele für Programme, die Offline arbeiten, sind MathMLStudio und tth.

Im folgenden werden einige Beispiele für Online-Konverter kurz dargestellt.
http://pear.math.pitt.edu/mathzilla/itex2mmlFrag.html bietet einen Itexx- Converter, anhand dessen zunächst die sich aus dem MathML- Code ergebende Darstellung angesehen werden kann, indem man das Fenster leert und die die eigene Formel einfügt. So kann man zunächst überprüfen, ob die tatsächliche Darstellung in MathML der Gewünschten entspricht. Anschließend kann man sich dort den MathML- Code erzeugen lassen, welcher dann über kopieren und einfügen in den HTML- Code der Webseite eingefügt werden. Vorteilhaft ist hier, dass der LaTeX- Code vorab daraufhin geprüft werden kann, ob die Ausgabe in einem HTML-Dokument wie gewünscht erfolgt.



In das Eingabefeld oben auf der Seite wird der LaTeX- Code eingegeben. Dann wird Die Auswahl „Display MathML“ darunter mit „Send text to be processed“ bestätigt. Danach erscheint im unteren Ausgabefenster die Ausgabe von MathML, wie oben zu sehen. Entspricht diese nicht den Vorstellungen, können weitere Formatierungen in LaTeX vorgenommen und der Vorgang wiederholt werden. Ist die Darstellung gelungen, kann durch Änderung der Auswahl auf „Show MathML code“ der Quelltext erzeugt und angezeigt werden, welcher dann in HTML eingebunden werden kann. Allerdings besteht hier keine Möglichkeit, eigene Dateien direkt upzuloaden und umzuwandeln.

Eine weitere Möglichkeit unter http://www.orcca.on.ca/MathML/texmml/textomml.html funktioniert sehr ähnlich, auch hier kann der LaTeX- Code nach dem Klicken auf „Clear“ in ein Fenster eingefügt werden, oder hier auch alternativ die URL des eigenen Dokuments angegeben werden, und sich dann die MathML-Ausgabe anzuschauen (Durch Klick auf „Translate“ am Seitenende).

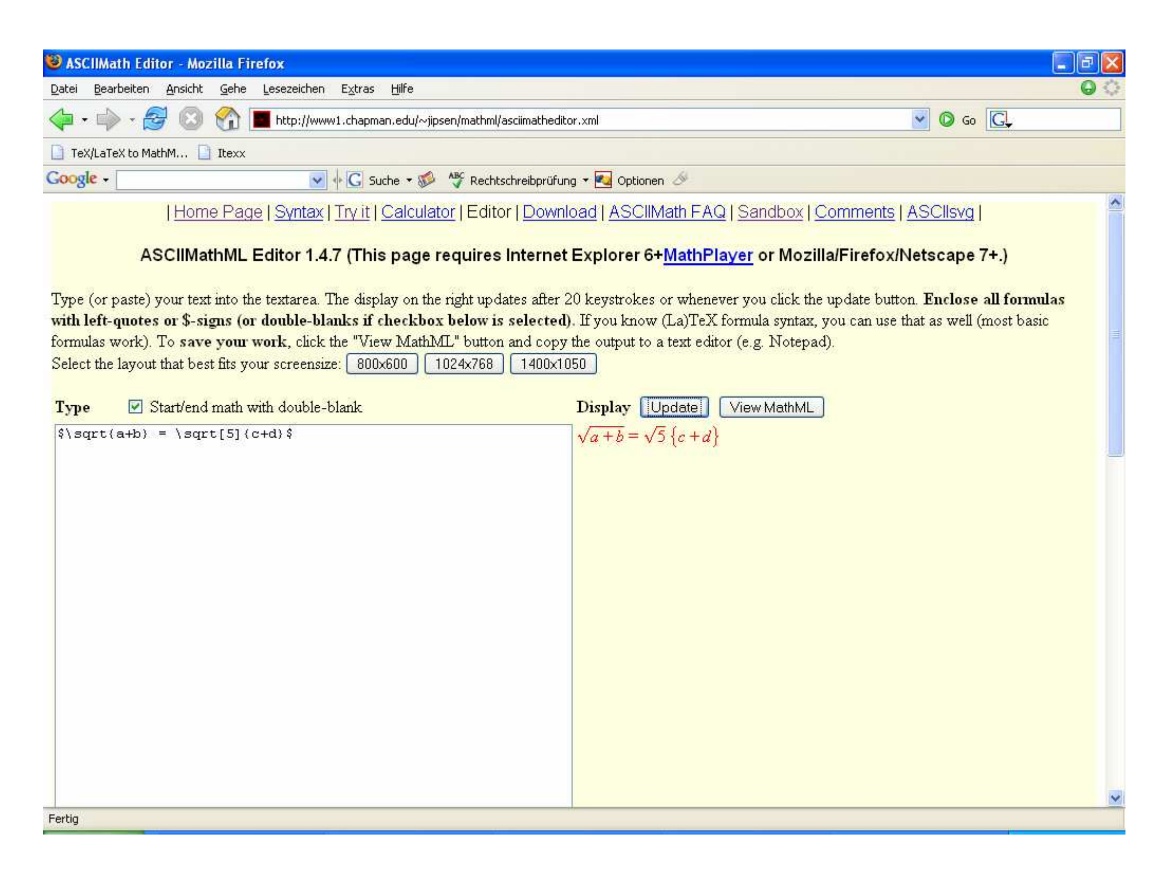


Die Darstellung erfolgt dann in einem neuen Fenster, das sich öffnet. Lässt man sich dann über „Ansicht“ in der Menüleiste den Seitenquelltext anzeigen, kann man diesen ebenfalls kopieren und in eigene Dokumente einfügen

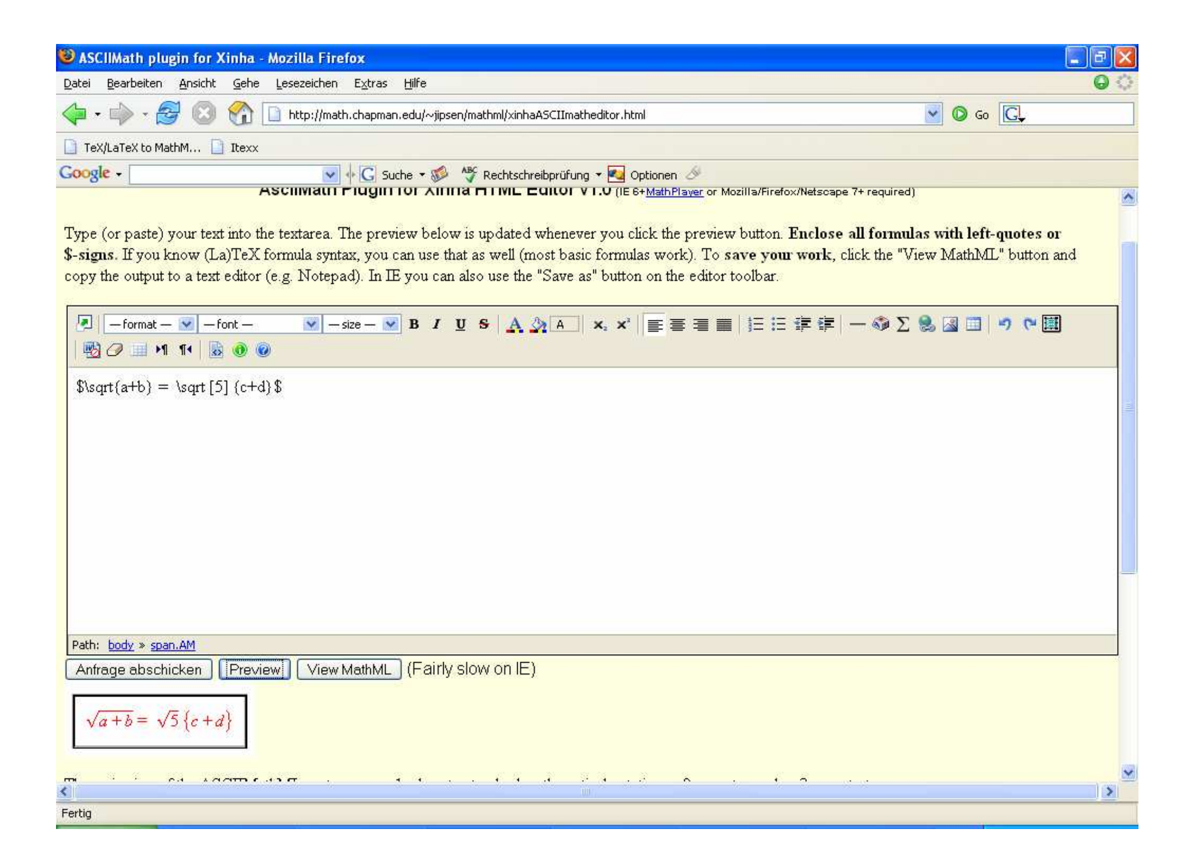
Die Möglichkeit, Dateien direkt uploaden zu können, stellt insbesondere bei komplexeren LaTeX-Formeln einen Vorteil im Hinblick auf den Komfort dar. Leider wird der MathML-Quelltext nicht direkt ausgegeben, man muss ihn extra aufrufen.

Eine dritte Möglichkeit, die Dateien online zu konvertieren, findet sich in Form des ASCII-MathML- Editors in der XML-Version der Seite unter http://www1.chapman.edu/~jipsen/mathml/asciimatheditor.xml oder in der HTML-Version der Seite unter mit dem Namen AsciiMath Plugin for Xinha HTML Editor unter http://math.chapman.edu/~jipsen/mathml/xinhaASCIImatheditor.html.

Diese arbeiten, anders als die Lösung vorher (Java), auf Basis von JavaScript, allerdings ist die Übersetzung in der XML-Version ähnlich gestaltet. Wenn man seinen LaTeX-Code ein-fügt, kann man sich die Ausgabe anzeigen lassen (Klick auf „Update“ oder aber direkt den MathML- Code erzeugen lassen (Klick auf „View MathML“). Ein direkter Upload der Datei ist leider nicht möglich.

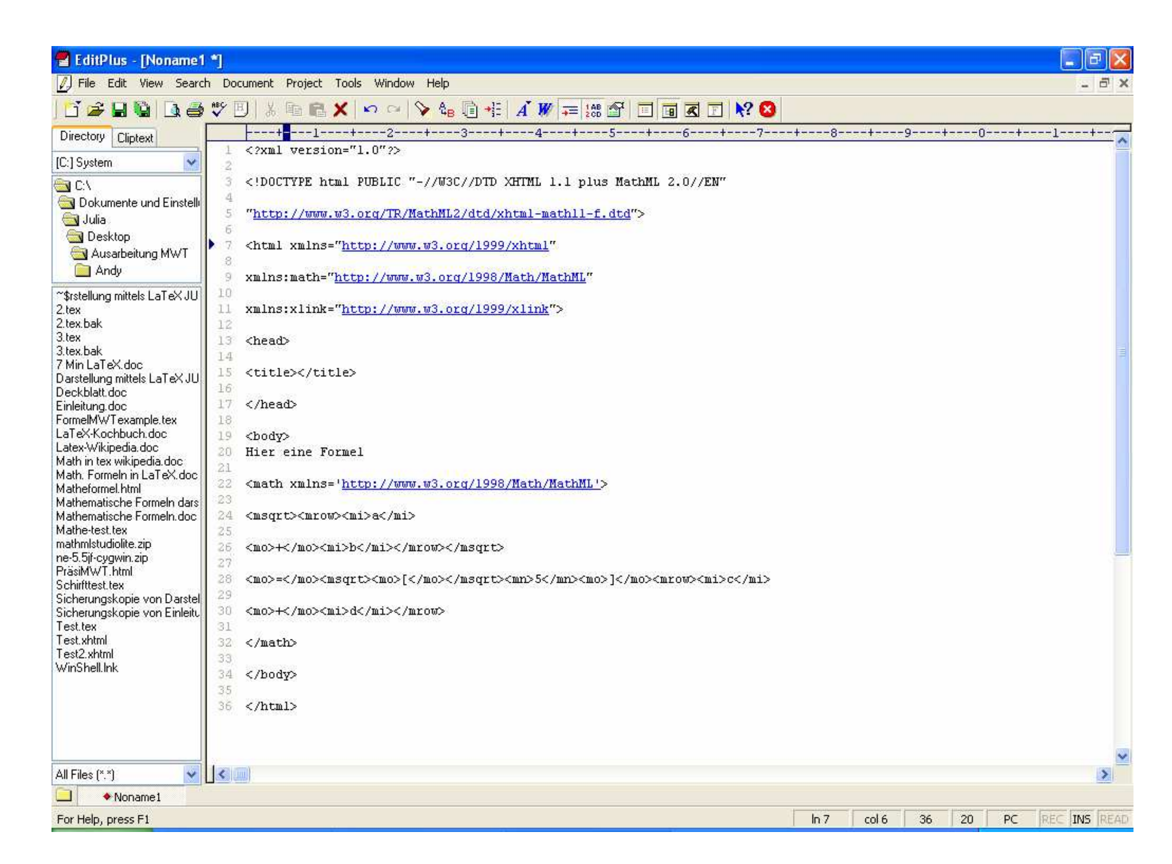


Die HTML-Version weicht davon etwas ab. Wie in der nächsten Abbildung zu erkennen, finden sich hier umfangreiche Möglichkeiten, um Formeln in LaTeX direkt im Fenster zu editieren (z.B. Darstellung in Fettschrift, erzeugen von Formeln ähnlich wie im Formeleditor anhand von Symbolen). Die Vorschau über „Preview“ und die Ausgabe des MathML-Codes über „View MathML“ sind hier ebenfalls möglich.
Die Möglichkeit, Formeln ähnlich einfach wie in einem Formeleditor zu erstellen und dann umwandeln zu lassen, ist auf jeden Fall vorteilhaft, vor allem für Personen mit geringen Vorkenntnissen in LaTeX (Die Einfassung der Formeln in Dollarzeichen muss von Hand erfolgen, worauf aber auch hingewiesen wird). Obwohl lt. Angaben auf der Seite diesbezüglich möglich, ist es im Versuch nicht gelungen, die Formel per Copy und Paste einzufügen, was bedeutet, dass die Eingabe per Hand erfolgen muss. Daher ist dies bei bereits existierenden TeX-Dateien wenig sinnvoll, da umständlich. Ein Upload der Dateien ist nicht möglich.

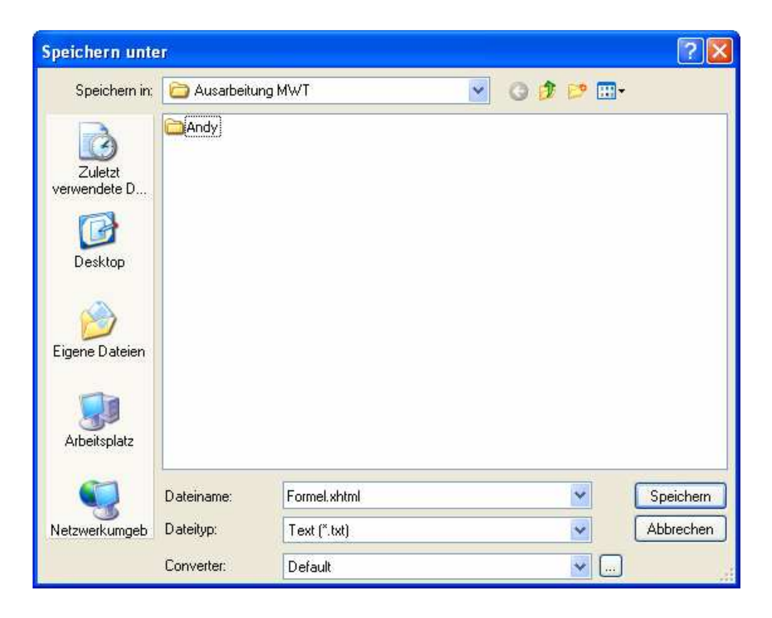


Welche dieser Lösungen die empfehlenswerteste ist, hängt zum einen vom Umfang LaTeX-Formel ab, aber auch von den Kenntnissen in LaTeX. Für umfangreiche Formeln ist mit Sicherheit die Möglichkeit des Dateiuploads interessant, wer über geringe Kenntnisse verfügt, für den ist sicherlich die Möglichkeit des Editierens von Vorteil.

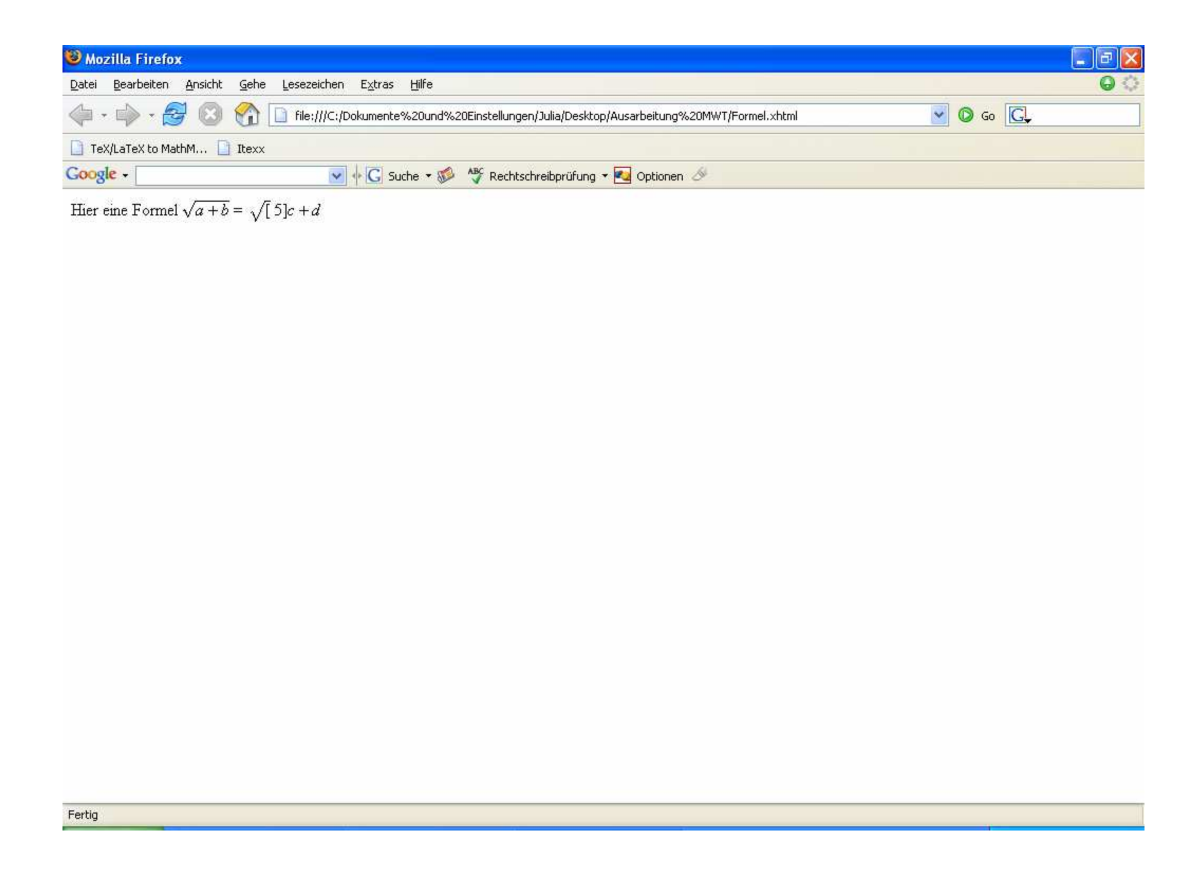
Nachdem wie oben beschrieben die Umwandlung in MathML durchgeführt wurde, muss der Quelltext in ein entsprechendes Dokument eingefügt und gespeichert werden, um auf dem eigenen Rechner im Browser dargestellt werden zu können.
Zu diesem Zweck empfiehlt sich, wie bereits erwähnt, die Möglichkeit, den Quelltext nach der Umwandlung im Web zu markieren, und diesen dann mit Copy & Paste in den Editor einzufügen. Dazu eignet sich in EditPlus zum Beispiel eine leere Seite.
In der Datei können nun die weiteren Eingaben gemacht und die Seite erstellt werden.



Wichtig ist, dass die fertige Datei nicht als einfaches HTML-Dokument gespeichert werden darf, da sonst die Formeln im Browser nicht korrekt dargestellt wird. Deshalb ist es erforderlich, dass die Datei beim Speichern die Endung „.xhtml“ erhält.



Nachdem die obige Datei auf dem Rechner gespeichert wurde, kann Sie im Browser aufgerufen werden. Die Darstellung sieht wie folgt aus:



4.6 Fazit

Eine gewisse Einarbeitung ist nötig, um mathematische Formeln mit Hilfe von LaTeX darstellen zu können, vor allem, um sich alle Möglichkeiten dieses Schriftsatzsystems bewusst zu machen. Auch die Erstellung in LaTeX mit anschließender Umwandlung in MathML mag umständlich erscheinen, allerdings ist, wie am Beispiel zu erkennen, LaTeX wesentlich übersichtlicher und einfacher als MathML. Wer einige Besonderheiten, wie z.B. die Speicherung der Datei als XHTML beachtet, kann mit dieser Methode mathmatische Formeln relativ einfach und ansprechend darstellen.